

COMP.SE.100-EN ItSE
Zoom begins soon...
at 1415 o'clock.



COMP.SE.100-EN, 2020, course schedule v6c (02.09.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		WE2: Trellis and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		WE4: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	WE5: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		WE10: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT	EXAM-3	groups' final presentations	groups' final presentations / feedback g-to-g (PRP)	48
49	L13: recap, summary	EXAM-3	---	final (2.) delivery of project documentation	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	end of game / game over.	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10.		

Remote/distant learning 2020.
No contact teaching at ItSE 2020.



COMP.SE.100 -EN "ItSE"

Introduction to Software Engineering

2020, 1-2. periods

5 credit units

05-diagrams-ItSE-2020-v6



COMP.SE.100-EN (ItSE)
Introduction to Software Engineering

Lecture 5, 30.09.2020

Tensu: remember to start Zoom
lecture recording, at 1415

Prefer course Moodle over SISU information.

Students are recommended to follow Moodle News/messages.

Instructions for students

Consider these at ItSE 2020;
WE, and PA 1st and final presentations.

- We recommend using a **profile picture**, which makes it easier to identify the speaker even without video.
- **Always blur the background of a video.** Avoid unnecessary visibility into the space where you are.
- If possible, stay in a space that others will not enter during the teaching sessions.
- **Tell your loved ones** in the same space that you are attending a distance learning session that may use live image.
- Make sure that no inappropriate or confidential material is visible in the **background**. If possible, look for a quiet place for the teaching session where the background is neutral.
- Mute additional audio sources in the distance learning situation.
- **Shut down the camera** if something irrelevant enters the camera's shooting range and during breaks.
- If you do not want to comment orally during video recording, use the chat window.

COMP.SE.100-EN, 2020, course schedule v6c (02.09.2020)

week	lectures	exam	weekly exercises	project assignment (exercise work)	week
35	L1: course basics		--- sign to WE groups ---	sign for project = grouping...	35
36	Project Assignment explained		WE1: intro to requirements	grouping, groups to Moodle	36
37	L2: Sw Eng in general		WE2: Trellis and agile way	group's Trello board ready with product backlog	37
38	L3: requirements		WE3: feasibility study and stakeholder analysis	working...	38
39	L4: basic UML diagrams		WE4: requirements	working...	39
40	L5: more UML diagrams	EXAM-1	WE5: UML diagrams - Use case	working...	40
41	L6: different sw systems	EXAM-1	WE6: UML diagrams - concept/entity and navigation	deadline for 1st phase documentation and presentation	41
42	examination week		examination week	examination week	42
43	L7: life cycle models		groups' 1st presentations	groups' 1st phase presentations	43
44	L8: quality and testing	EXAM-2	WE7: development processes	feedback group-to-group at PRP, from 1st phase	44
45	L9: project work	EXAM-2	WE8: testing and error reporting	deadline for diagrams first versions (Moodle)	45
46	L10: project management		WE9: effort estimation	feedback to groups from diagrams (from assistants)	46
47	L11: open source, APIs, IPR		WE10: delivery contracts and terms of use	deadline for 2nd phase presentation (PRP)	47
48	L12: embedded systems, IoT	EXAM-3	groups' final presentations	groups' final presentations / feedback g-to-g (PRP)	48
49	L13: recap, summary	EXAM-3	---	final (2.) delivery of project documentation	49
50	examination week		examination week	feedback inside group, student-to-student at PRP	50
51	examination week		examination week	end of game / game over.	51
	Lectures: Wed at 1415-16.		Weekly exercises:		
			Mon 0815-10	AUTUMN 2020 (1-2. periods)	
			Mon 1215-14	are remote/distant learning.	
			Tue 0815-10		
			Tue 1415-16		
			Wed 0815-10.		

First, general course matters

Juanita: groups G01-G04

Aleksius: ODD groups; G05,G07,G09,G11,G13,G15,G17,G19,G21,G23,G27

Lauri: EVEN groups; G06,G08,G10,G12,G14,G16,G18,G20,G22,G24,G28

- Trello board is used as help for work division and assignment

WE attendees:

- Mon 0815-10 9, 8,10, 5, 6,
- Mon 1215-14 11,12,12,13,11,
- Tue 0815-10 3, 6, 4, 6, 5,
- Tue 1415-16 8,10, 9, 8, 5,
- Wed 0815-10 12,11, 9, 8, 7,

Very small WEs are not reasonable,
we think how many groups will
continue at 2nd period...

Current at course (w 40)

- WE5 was about Dia use, and Use cases
 - WE6 is about other useful diagrams needed at PA and EXAM
-
- EXAM 1/3 (w40-41) is ongoing, 5 taken, 20 reservations so far, do not leave your EXAM reservation for the last days
-
- for WE5 and WE6, install Dia to your own computer
 - (EXAM classes have Dia and EA tools installed)
-
- remember to use (i.e. update) your group's Trello board.

Course contents (plan)

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
4. Basic UML Diagrams ("Class", Use Case, Navigation)
5. **UML diagrams, in more detail**
6. Different software systems
7. Life Cycle models
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems
13. Recap

5. UML diagrams, in more detail

- Just a few more UML diagrams...
 - **state machine diagrams** ("state transition diagrams", "state diagrams", "state machines")
 - **sequence diagrams** ("scenarios")
 - **activity diagrams**
 - activity diagrams **with partitions** ("swimlane diagrams")
 - flowchart
 - Gantt chart

And once again, naming of the diagrams may be different "at streets and in companies".

Some other useful UML diagrams [uml-diagrams.org]

state machine diagrams ("state transition diagrams", "state diagrams", "state machines")

- Used for modeling discrete behavior through finite state transitions. In addition to expressing the behavior of a part of the system, state machines can also be used to express the usage protocol of part of a system. These two kinds of state machines are referred to as behavioral state machines and protocol state machines.

sequence diagrams ("scenarios")

- most common kind of interaction diagrams which focuses on the message interchange between lifelines (objects).

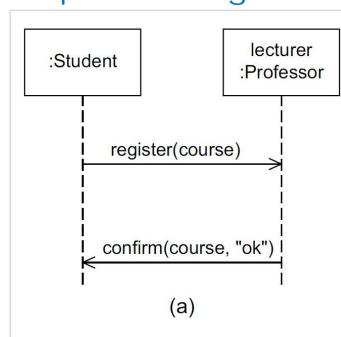
activity diagrams

- Shows sequence and conditions for coordinating lower-level behaviors, rather than which classifiers own those behaviors. These are commonly called control flow and object flow models.

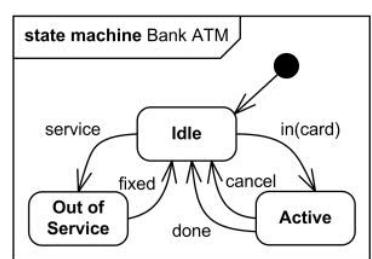
activity diagrams with partitions ("swimlane diagrams").

Some other useful UML diagrams [uml-diagrams.org]

sequence diagram

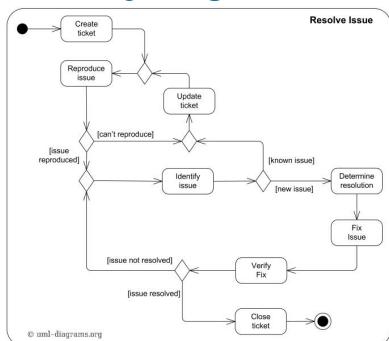


state machine

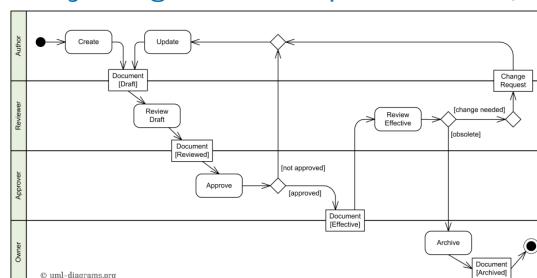


All diagrams are made to help software engineering work.

activity diagram



activity diagram with partitions ("swimlane diagrams").



Basic concepts of ... software development

Balzert vol. 1, 2nd edition 2001

Yes, there are many and many different kind of methods and diagramming techniques... for example classified by view, just pick the most useful to your specific project and use them.

Concepts and Views											
Alternative Notations											
Often used											
Rarely used											
Function tree	Use Case Diagram 1987	Data flow diagram 1966	Data-Dictio-nary 1979	Entity Rela-tionship Model 1976	Class diagram 1980/ 1990	Pseudo code	Box diagram 1973	Decision tables 1957	Activity diagram 1997	Collabo-ration diagram	
Functional hierarchy	Business Process	Information Flow	Data Structures	Entity types and relations	Class structures	Control structures	If-Then structures	State automaton 1954	Petri Net 1962	Sequ-ence diagram 1987	
Functional View		Data-Oriented View		Object-Oriented View	Algorithmic View	Rule-Based View	State-Oriented View		Scenario-Based View		

DAAD project „Joint Course on Software Engineering“ ©

5

TUNI / COMP.SE.100-EN

30.09.2020 10.03

13

UML 2.5.1 (2017) specification

www.omg.org/spec/UML/2.5.1/PDF

An OMG® Unified Modeling Language® Publication




OMG® Unified Modeling Language® (OMG UML®)

Version 2.5.1

OMG Document Number: formal/2017-12-05

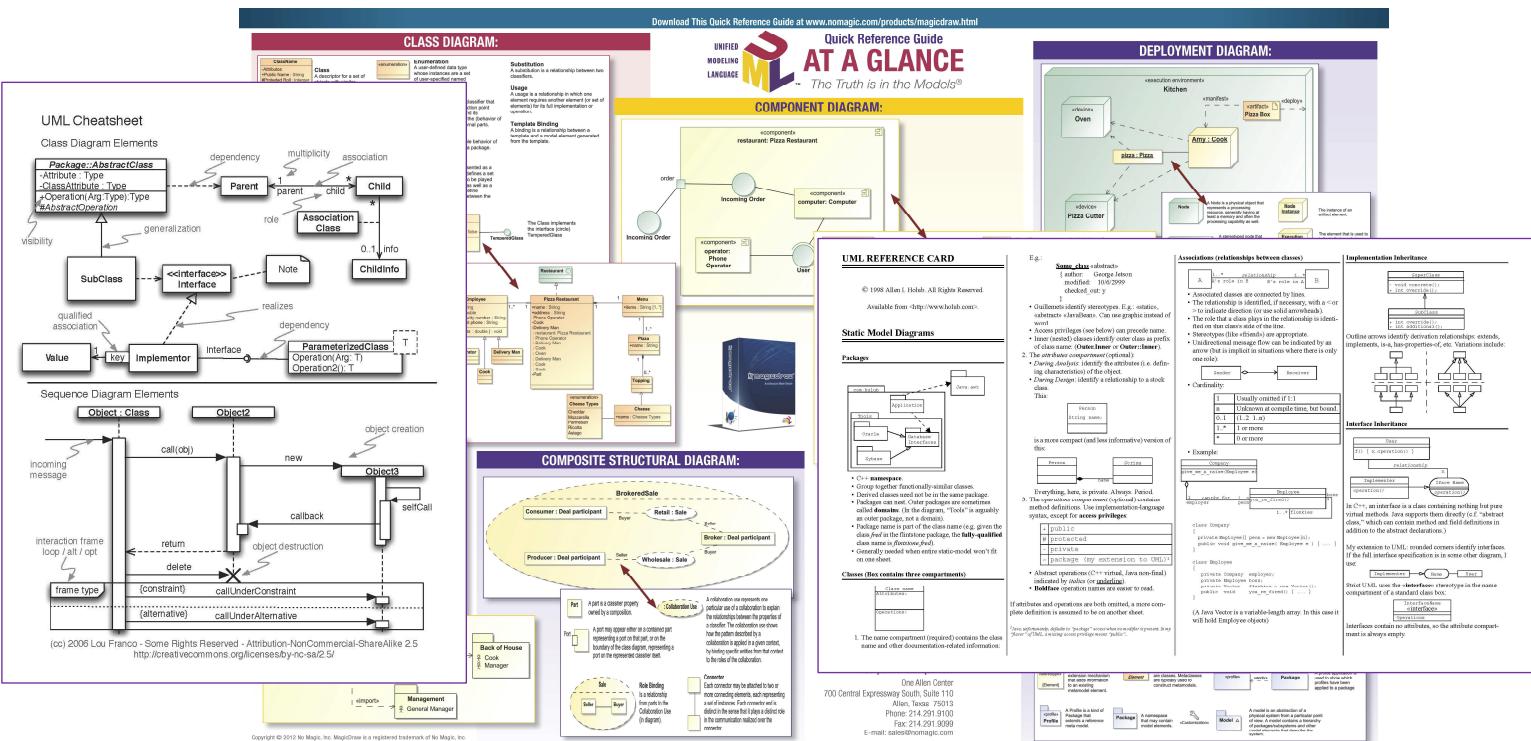
Date: December 2017

Normative URL: <https://www.omg.org/spec/UML/>

Machine Readable:

Normative: <https://www.omg.org/spec/UML/20161101/PrimitiveTypes.xmi>
<https://www.omg.org/spec/UML/20161101/UML.xmi>
<https://www.omg.org/spec/UML/20161101/StandardProfile.xmi>
<https://www.omg.org/spec/UML/20161101/UMLDI.xmi>

There are many UML reference cards and cheat sheets available



TUNI / COMP.SE.100-EN

30.09.2020 10.03

15

UML 2.5.1 (2017) specification

www.omg.org/spec/UML/2.5.1/PDF

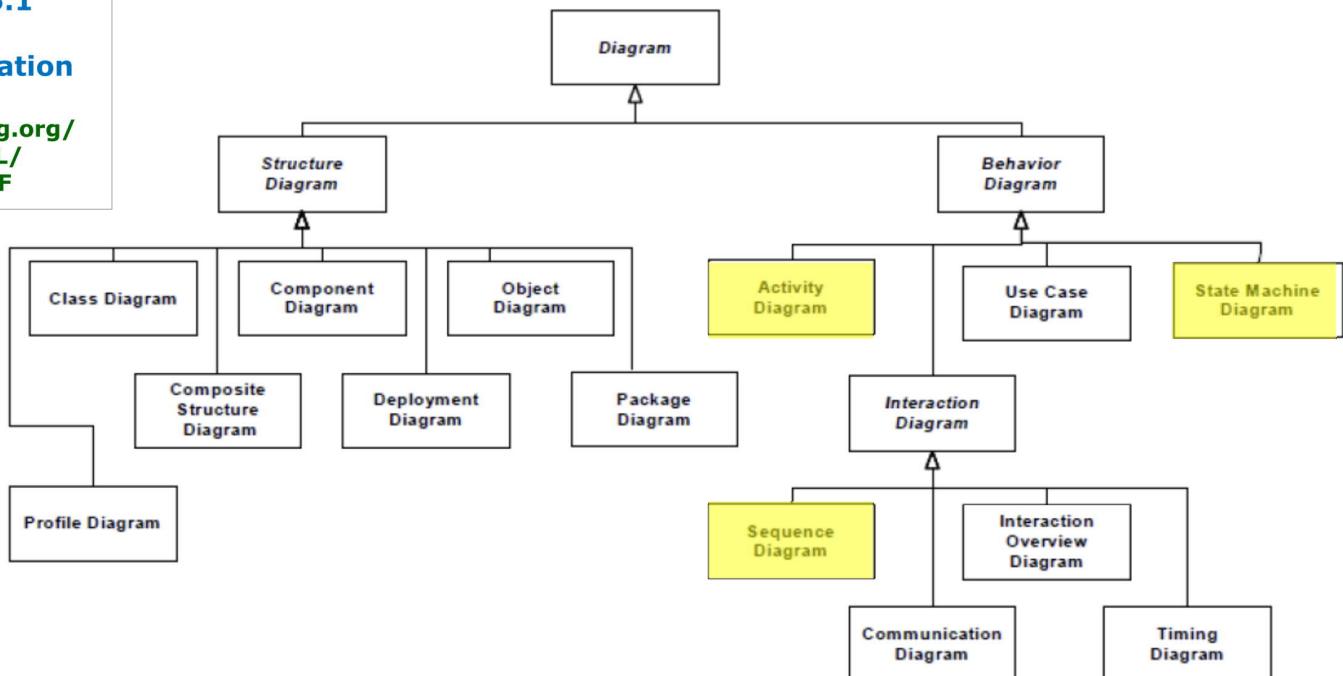


Figure A.5 The taxonomy of structure and behavior diagrams

TUNI / COMP.SE.100-EN

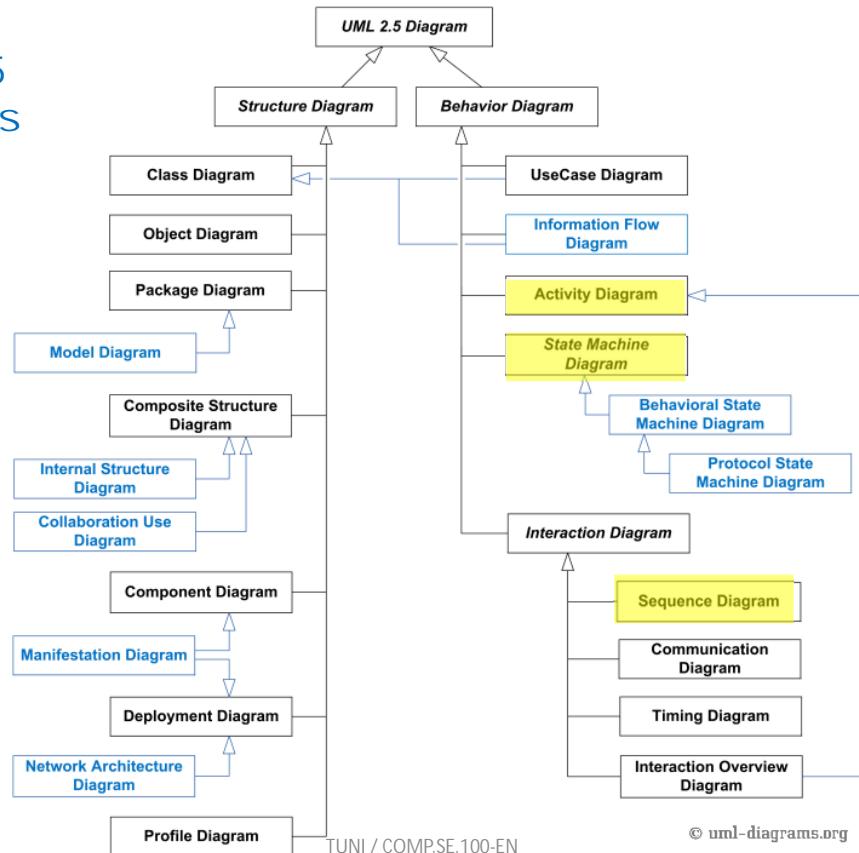
30.09.2020 10.03

16

UML 2.5 diagrams (2015)

UML diagram types

www.uml-diagrams.org/



30.09.2020 10.03

TUNI / COMP.SE.100-EN

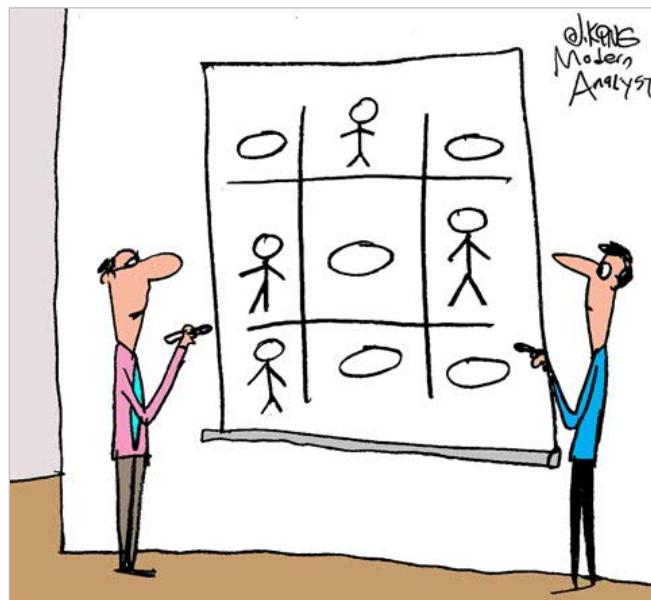
© uml-diagrams.org

17



Feel free to use whatever kind of useful diagrams and drawings in your future software engineering projects.

But remember then to explain the symbols and techniques to the other parties.

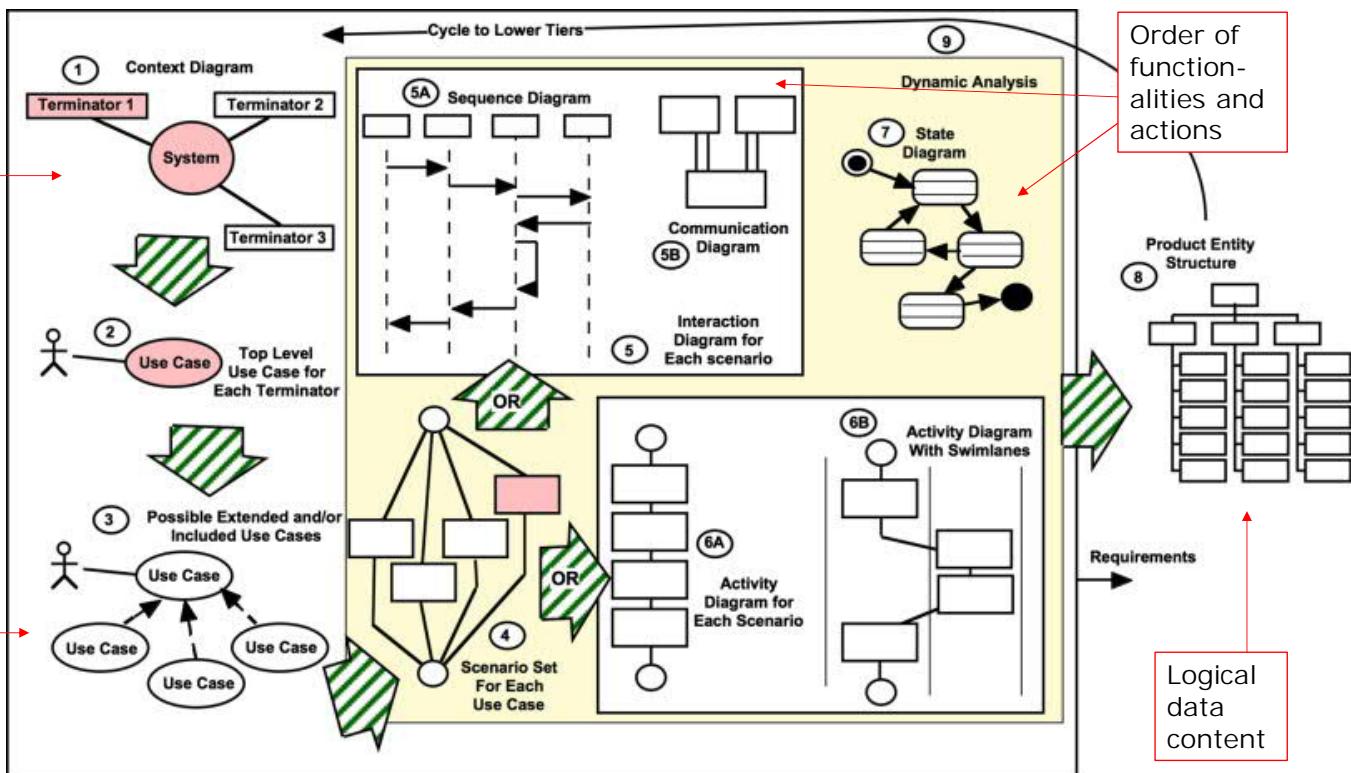


Jerry and Bob, the new business analysts, decided to settle, once for all, whether use cases or actors are more important in the use case diagram.

TUNI / COMP.SE.100-EN

30.09.2020 10.03

18



[ars.els-cdn.com]

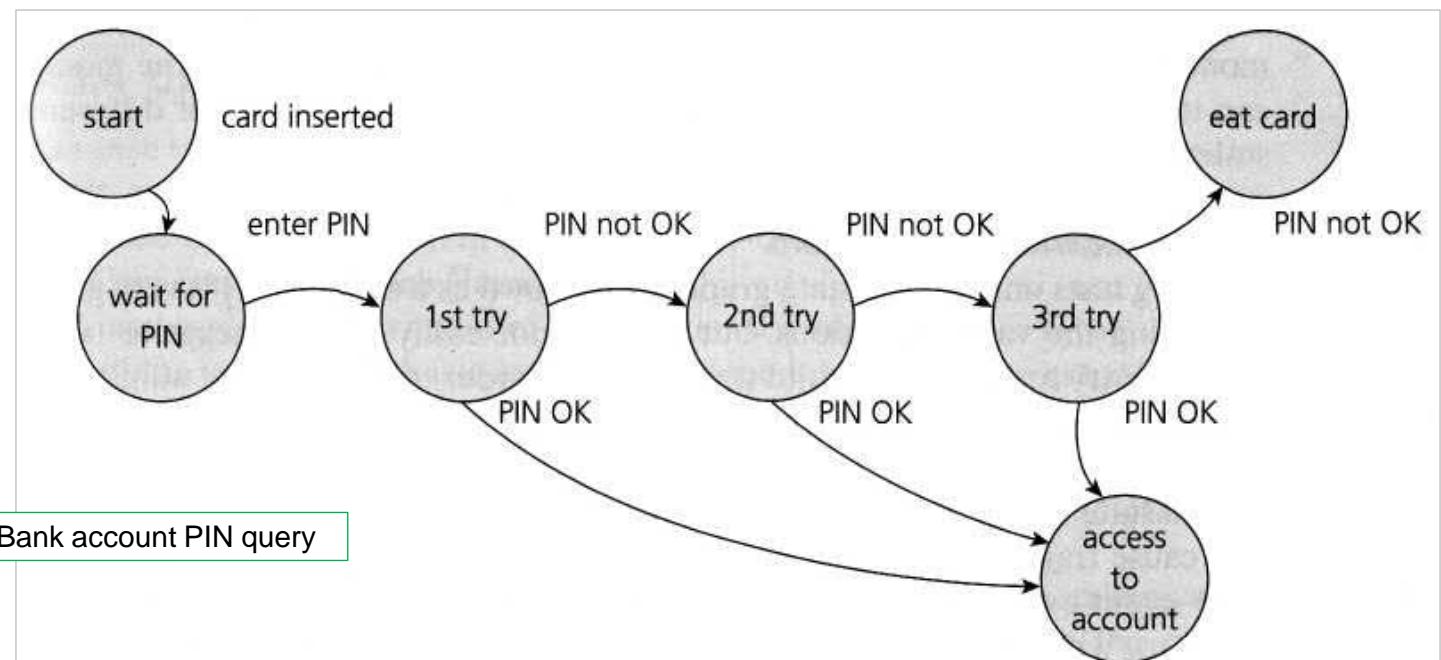
TUNI / COMP.SE.100-EN

30.09.2020 10.03

19

State transition diagrams = State machines

Basic state diagram, may be used early in requirements specification phase



Use of state diagrams

State diagrams may be used in "general" way in requirements phase, and in more detail in design phase when defining software architecture and to help developers to understand software's wanted behaviour in detail.

Sometimes state transition table (matrix) would help. From such it is easy to see all state transitions which are enabled. Sometimes it would be necessary in code to make sure that certain state transitions do not happen.

For example in some real-time systems it is very important in which order the functions are done.

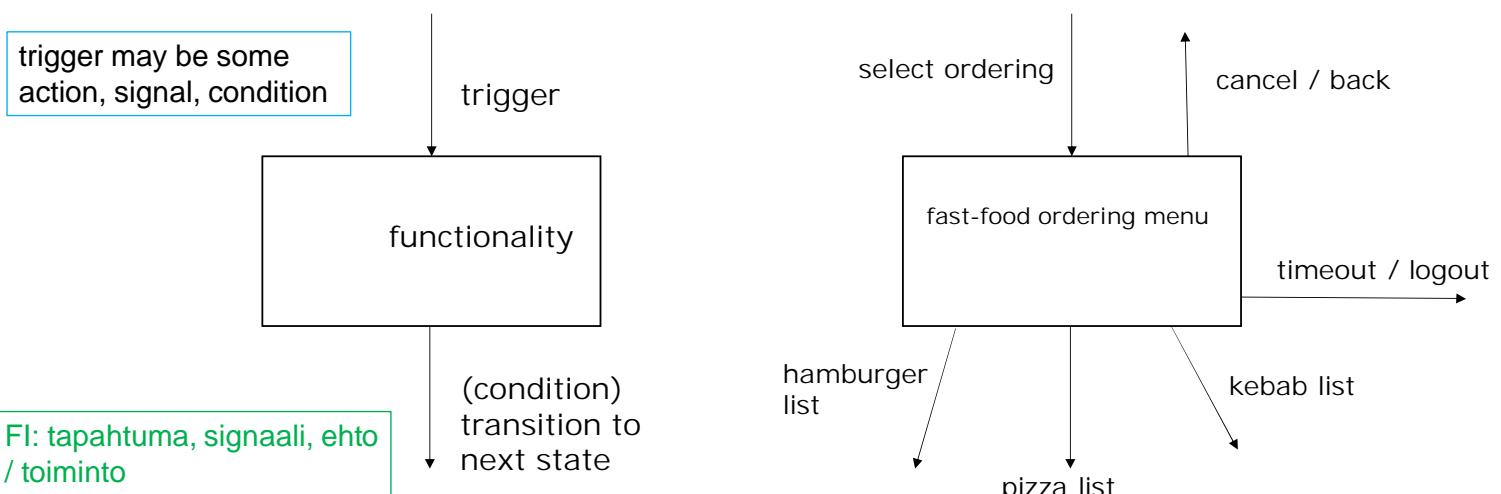
It depends also from stakeholders which kind of diagrams you use.

Why state (transition) diagrams

- The main idea is to show in which order operations are made or should happen in a program.
- What user must do first, before something else ?
- What triggers a transition (some action, time, condition, or is some state just a "go-through" state for some other purpose) ?
- Start and end should be marked clearly.
- Sometimes it is worth documenting what state transitions should never happen; you can not go from state X to state Z.
- State transition matrix may be useful sometimes.

State (transition) diagram, state machine (FI: tila(siirtymä)kaavio, tilakone)

Traditionally software's order of functionality (e.g. user's or processes' possible action paths) has been expressed by state transitions diagrams.



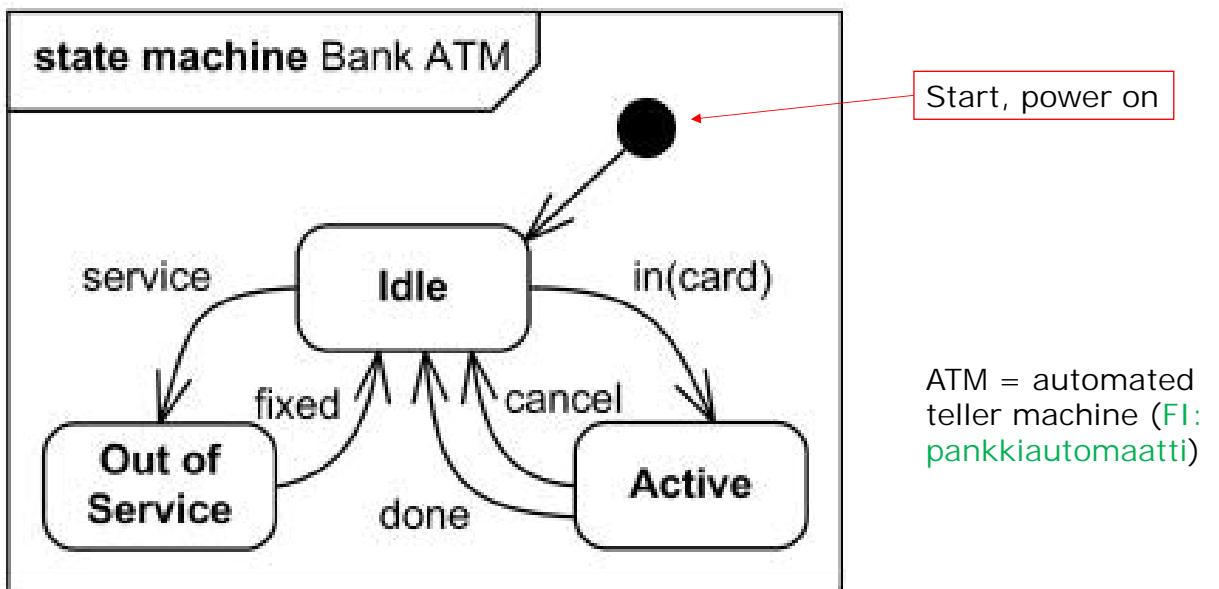
State machines [UML 2.5.1 spec]

The StateMachines package defines a set of concepts that can be used for modeling discrete event-driven Behaviors using a finite state-machine formalism. In addition to expressing the Behavior of parts of a system (e.g., the Behavior of Classifier instances), state machines can also be used to express the valid interaction sequences, called protocols, for parts of a system. These two kinds of StateMachines are referred to as behavior state machines and protocol state machines respectively.

Behavior StateMachines can be used to specify any of the following:

- The classifierBehavior of an active Class.
- An ownedBehavior of a BehavioredClassifier that is not the classifierBehavior of that BehavioredClassifier.
- A stand-alone Behavior, that is, one that does not have a corresponding BehavioredClassifier.
- A method corresponding to a BehavioralFeature (i.e., an Operation or a Reception).

[uml-diagrams.org]

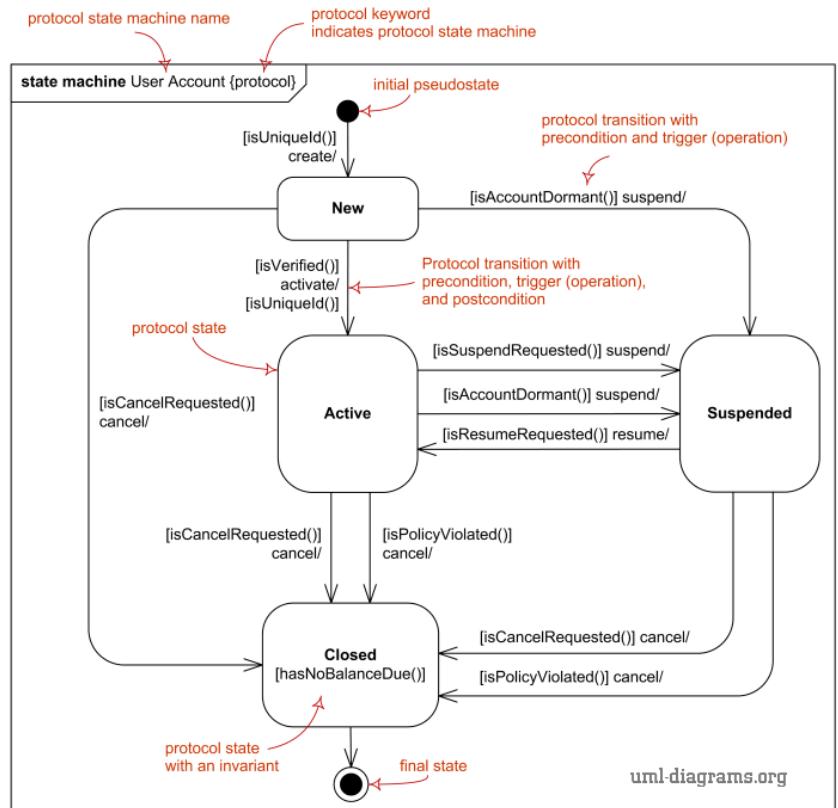


High level behavioral state machine for bank ATM.

UML **protocol state machine** diagrams are used to express a usage protocol or a lifecycle of some classifier. It shows which operations of the classifier may be called in each state of the classifier, under which specific conditions, and satisfying some optional postconditions after the classifier transitions to a target state.

State transition:

[preconditon] trigger / [postcondition]



State machine [UML specification, 2015]

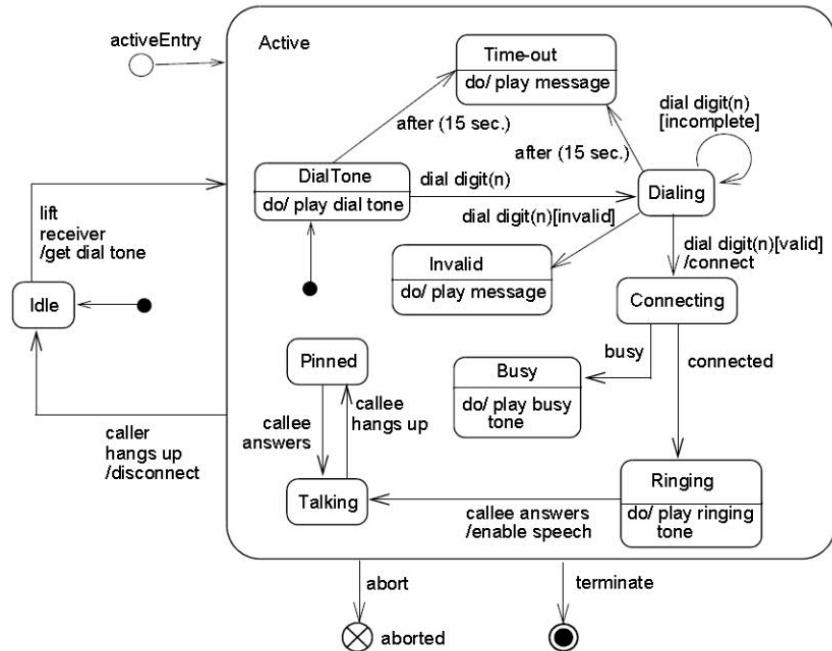
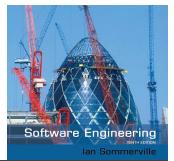
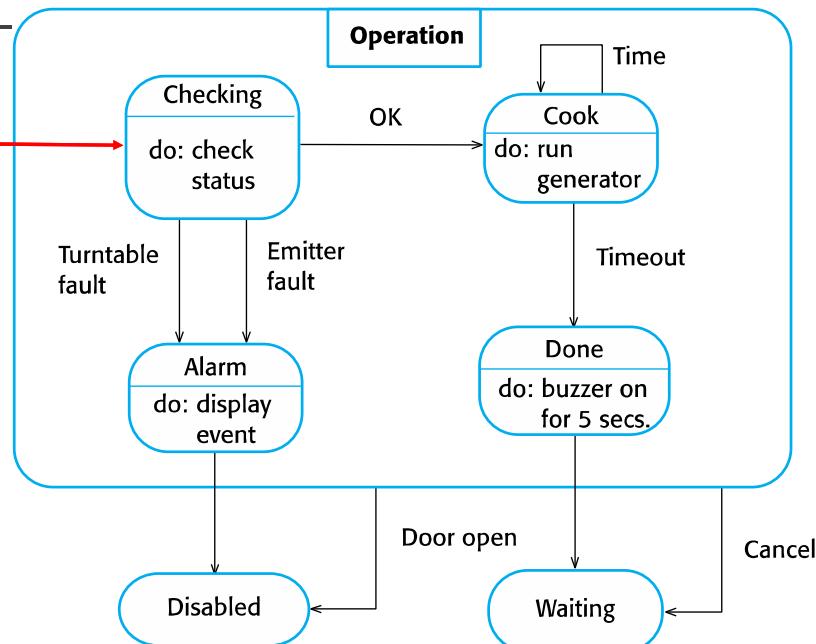


Figure 14.36 StateMachine diagram representing a telephone

Microwave oven operation



It is important to define from which state operating begins, when the device is turned on (also after power outage). So it will not go to some random state.



30.09.2020 10.03

TUNI / COMP.SE.100-EN

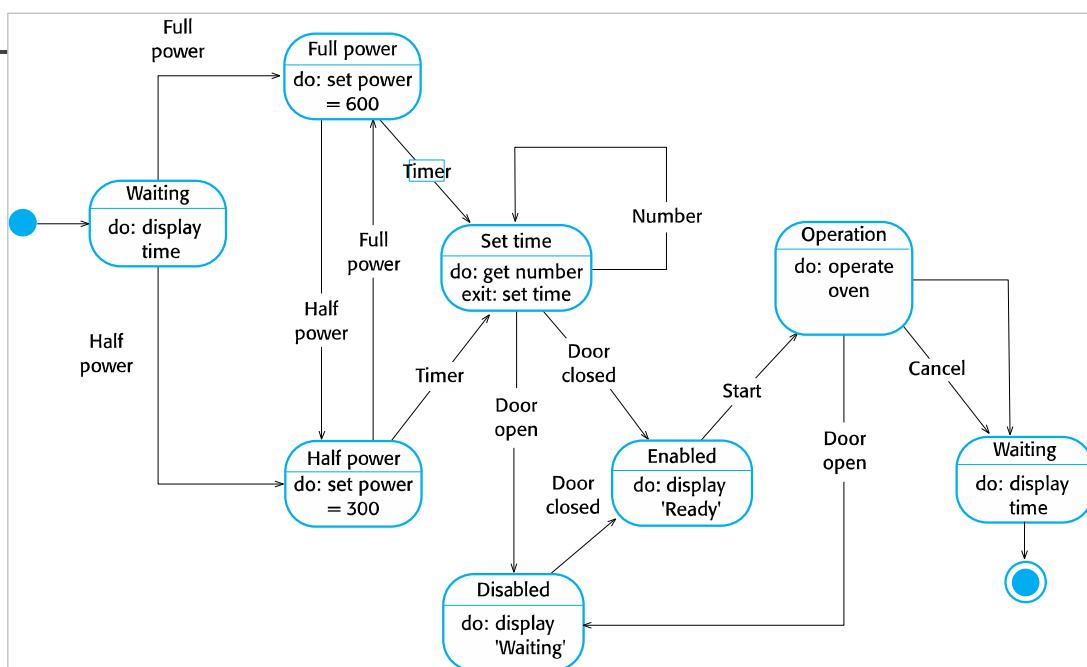
29

State diagram of a microwave oven



start

end



30.09.2020 10.03

TUNI / COMP.SE.100-EN

30

States and stimuli for the microwave oven



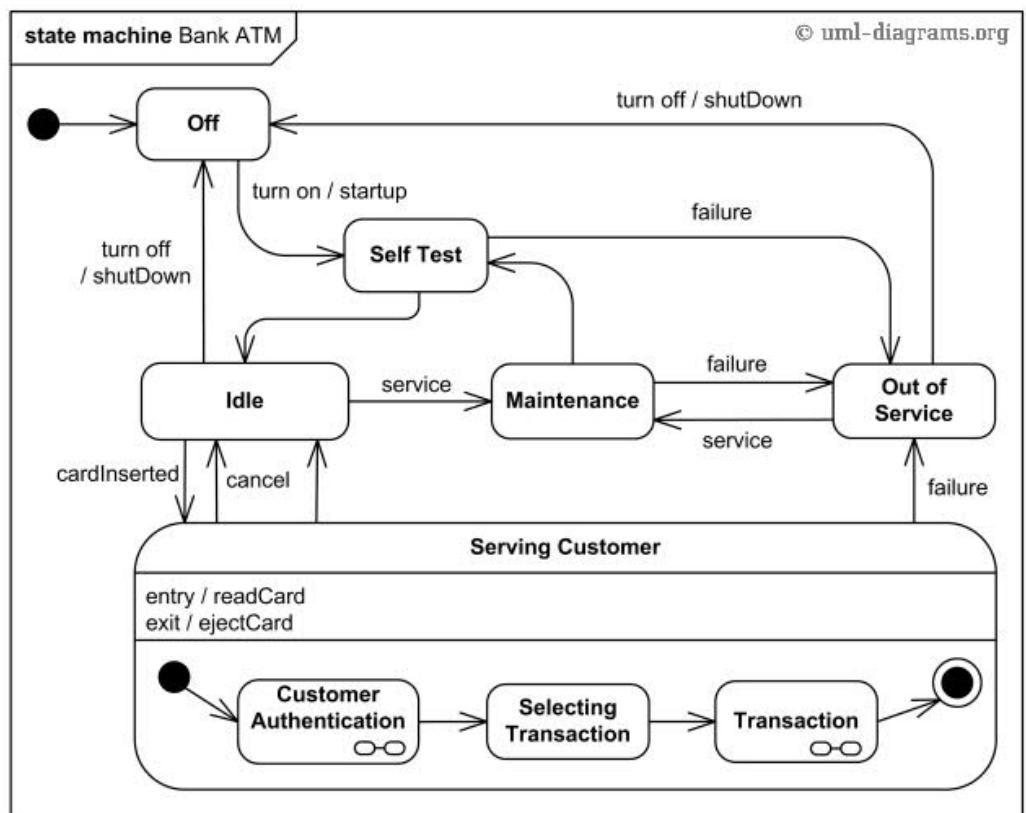
State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

30.09.2020 10.03

TUNI / COMP.SE.100-EN

31

Bank Automated
Teller Machine
(ATM) top level
state machine.



Two different diagrams (styles) of the same system, e.g. a web shop system

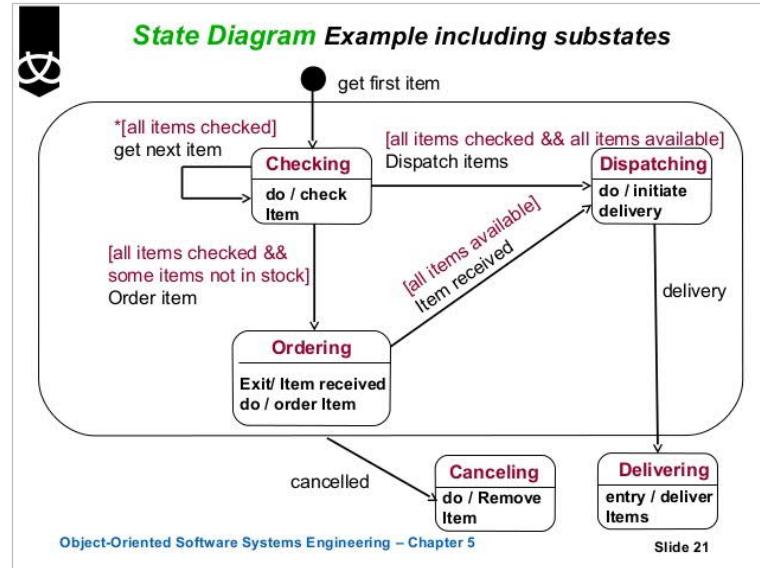
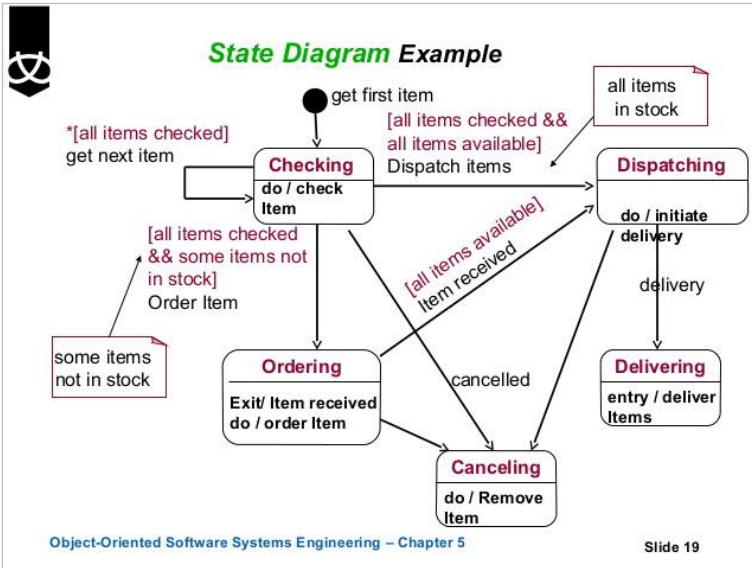


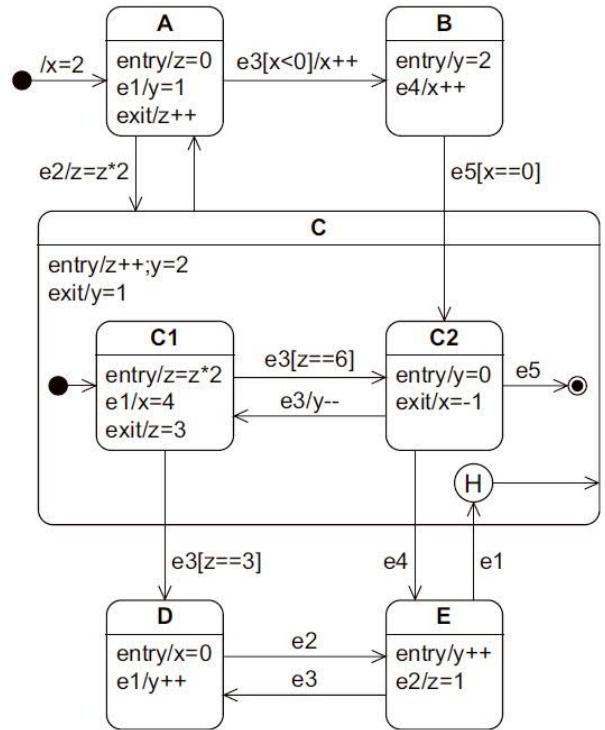
Figure 5.20
State machine diagram to demonstrate a sequence of events

State machine diagram

One kind of state table

Event	State entered	x	y	z
Start	A	2		0
e2	C1		2	6
e1	C1	4		
e3	C2		0	3
e4	E	-1	2	
e1	C2		0	4
e5	A	-1	1	0

Table 5.1
State changes and variable assignments for x, y, and z after the occurrence of the individual events



State machine diagram

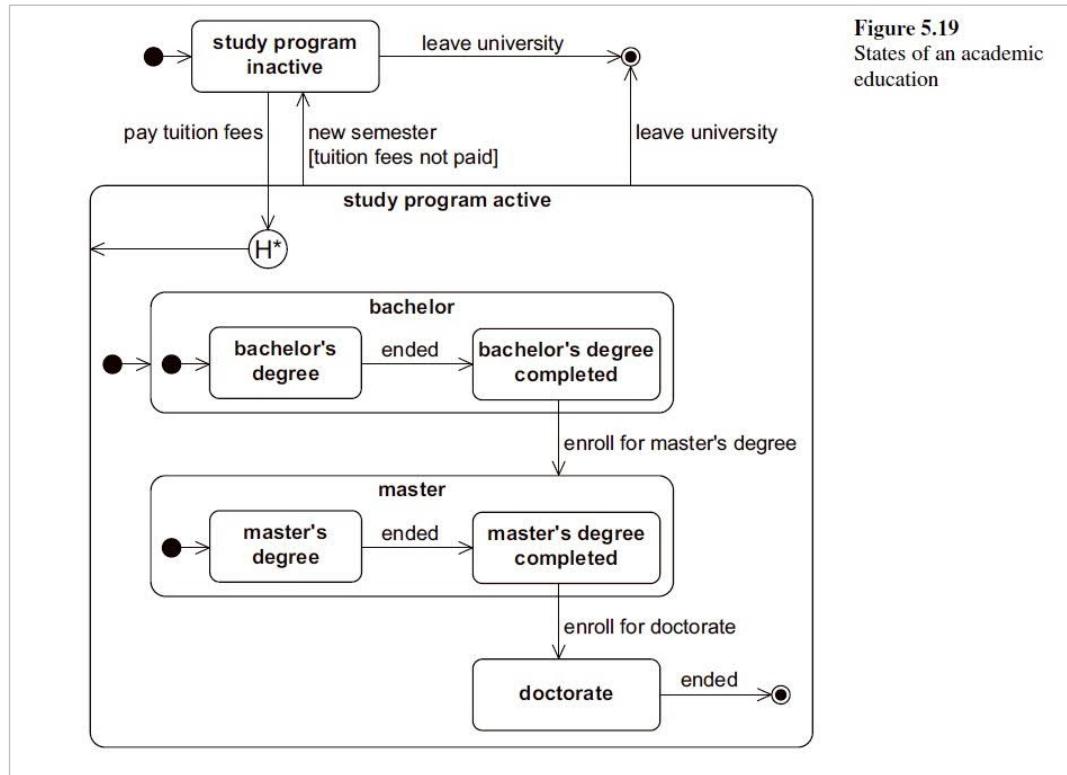


Figure 5.19
States of an academic education

TUNI / COMP.SE.100-EN

[UML@Classroom, 2015]

30.09.2020 13.49

35

Sequence diagrams

Sequence diagrams

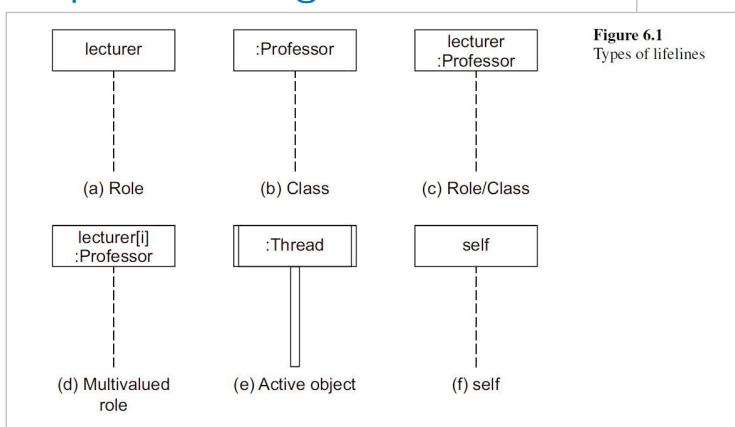
[<https://creately.com/blog/diagrams/sequence-diagram-tutorial/>]

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

In simpler words, a sequence diagram shows different parts of a system work in a 'sequence' to get something done.

A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.

Sequence diagram lifelines

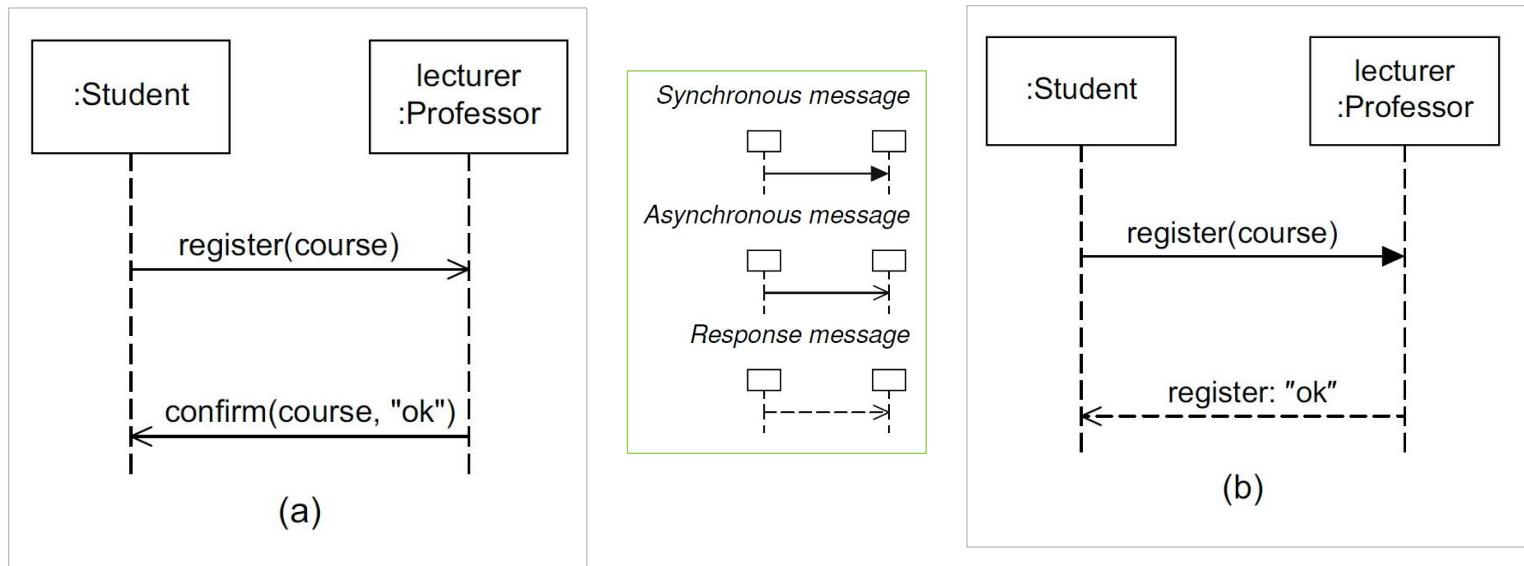


[UML@Classroom, 2015]

Table 6.2
Notation elements for the sequence diagram

Name	Notation	Description
Lifeline	 	Interaction partners involved in the communication
Destruction event		Time at which an interaction partner ceases to exist
Combined fragment		Control constructs
Synchronous message		Sender waits for a response message
Response message		Response to a synchronous message
Asynchronous message		Sender continues its own work after sending the asynchronous message
Lost message		Message to an unknown receiver
Found message		Message from an unknown sender

Sequence diagram, messages



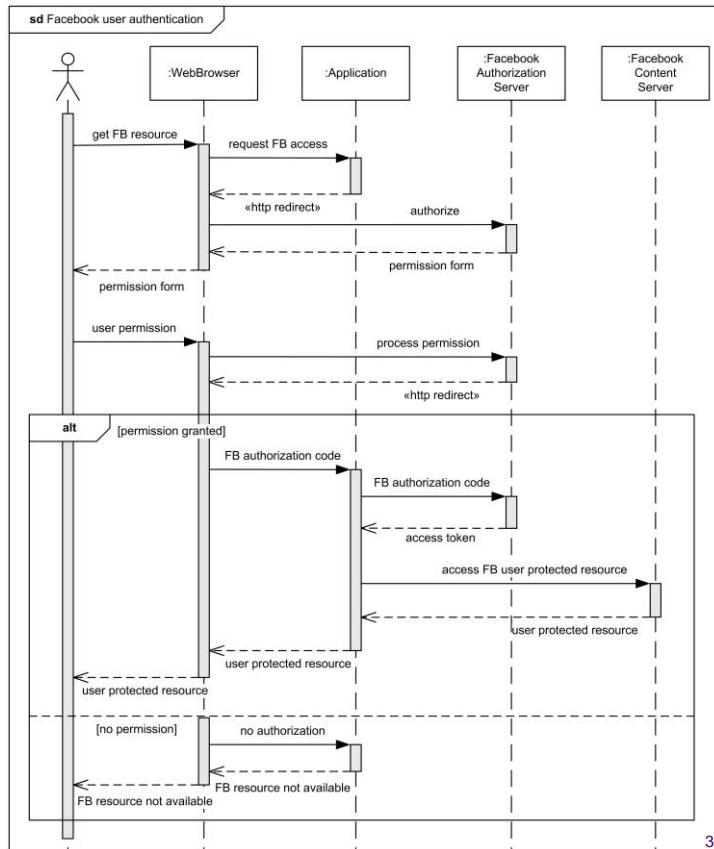
[UML@Classroom, 2015]

TUNI / COMP.SE.100-EN

30.09.2020 10.03

39

Sequence diagram;
 Facebook (FB) user could be authenticated in a web application to allow access to his/her FB resources. Facebook uses OAuth 2.0 protocol framework which enables web application (called "client"), which is usually not the FB resource owner but is acting on the FB user's behalf, to request access to resources controlled by the FB user and hosted by the FB server. Instead of using the FB user credentials to access protected resources, the web application obtains an access token.



TUNI / COMP.SE.100-EN

30.09.2020 10.03

40

Entity/Context diagram, or...

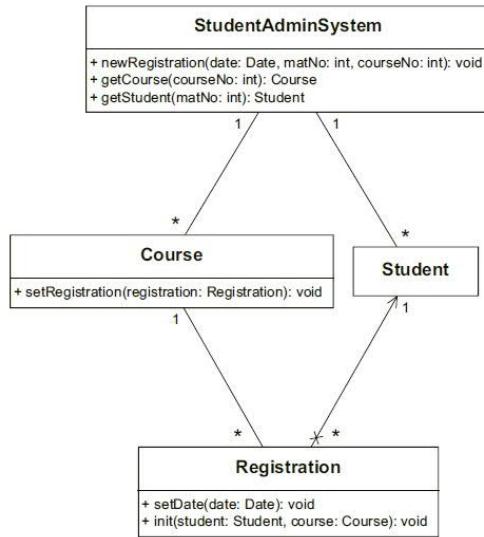
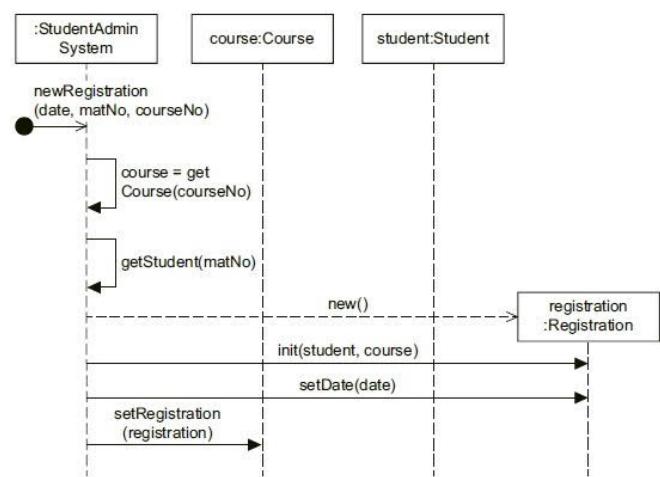


Figure 6.30
Sequence diagram based on class diagram



Sequence diagram

[UML@Classroom, 2015]

COMP.SE.100-EN (ItSE) Introduction to Software Engineering

Lecture 5, 30.09.2020

Tensu: remember to pause
Zoom lecture recording

Zoom lecture break, 10 minutes stretching, walking, etc.

Activity diagrams

Activity diagrams

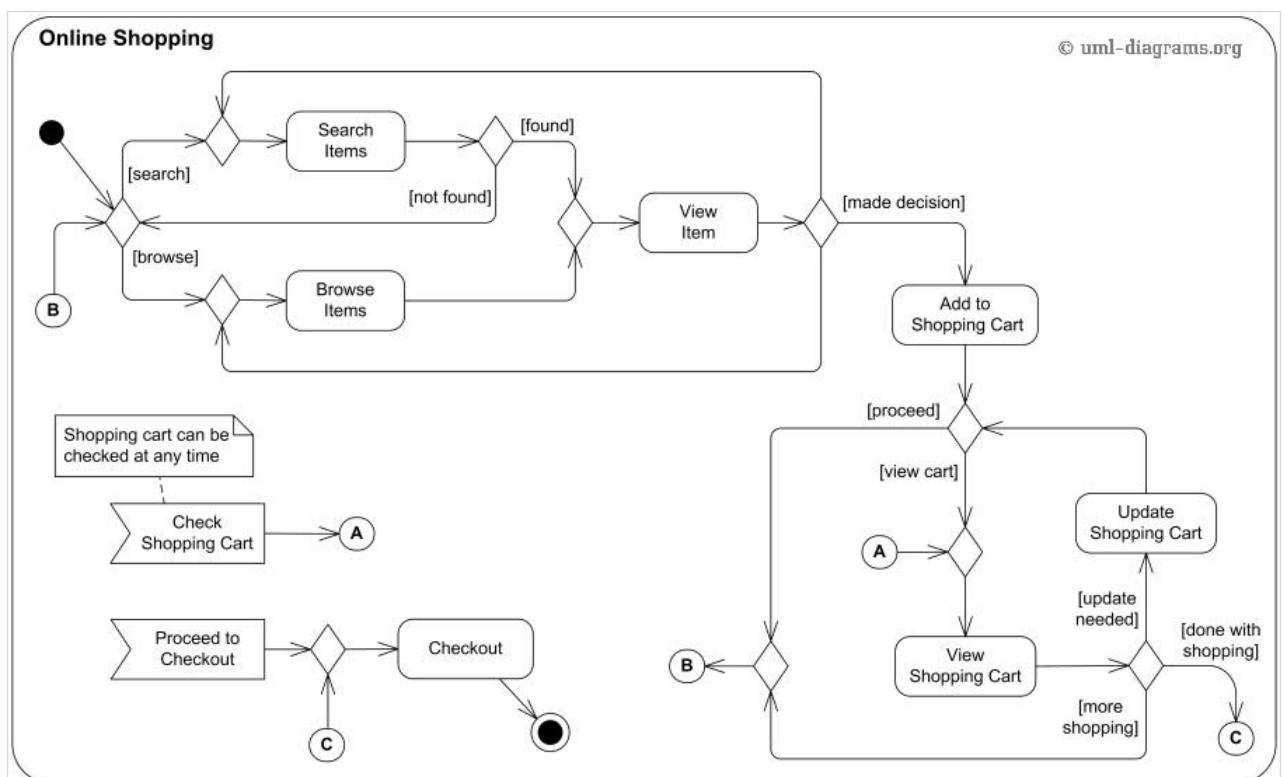
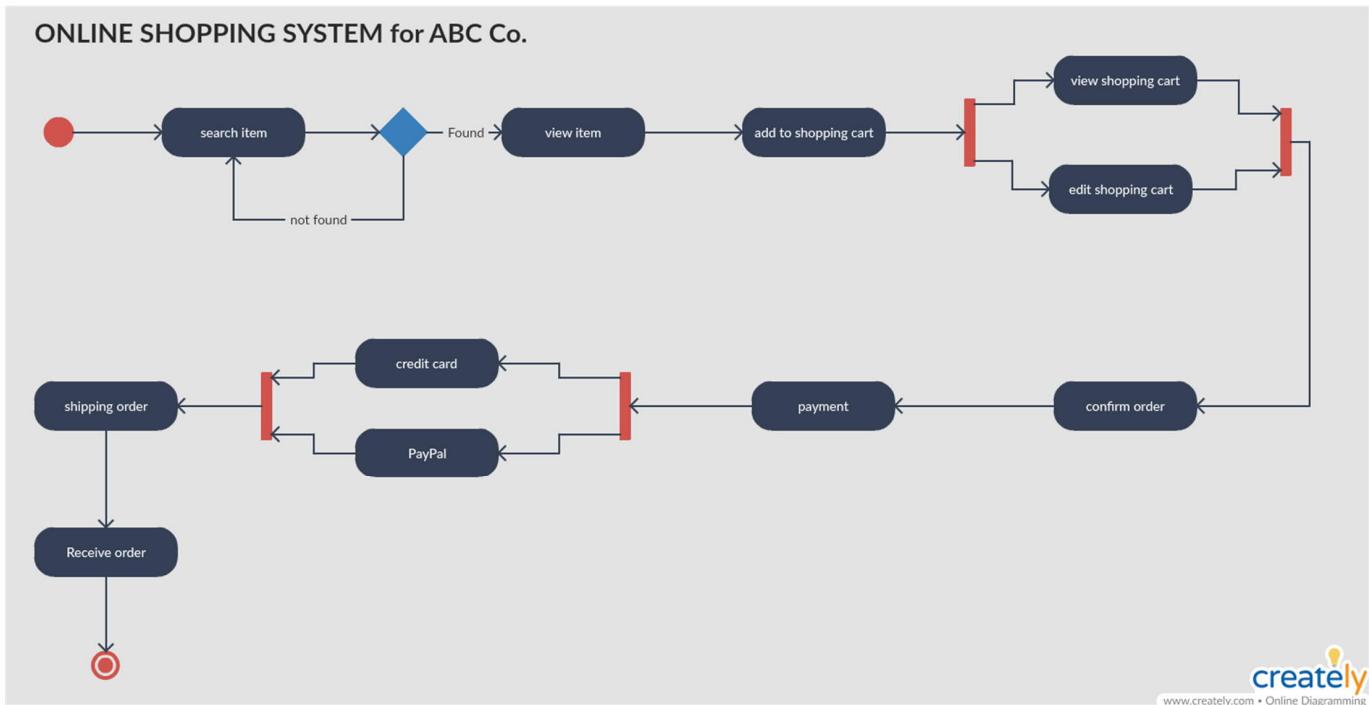
[<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/>]

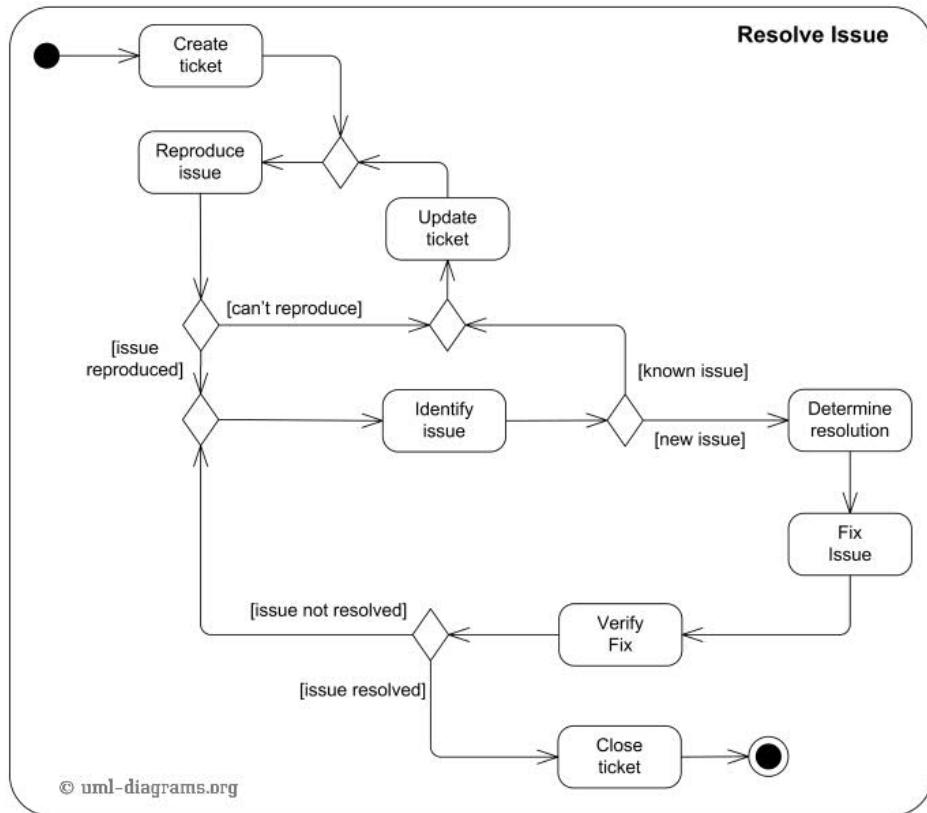
Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows

- Identify candidate use cases, through the examination of business workflows
- Identify pre- and post-conditions (the context) for use cases
- Model workflows between/within use cases
- Model complex workflows in operations on objects
- Model in detail complex activities in a high level activity Diagram.

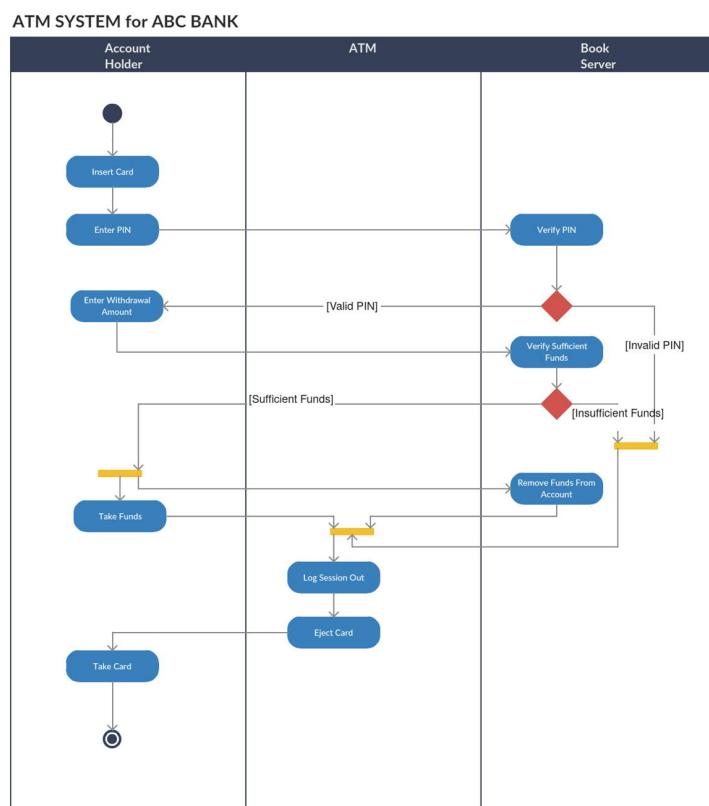
Activity diagram

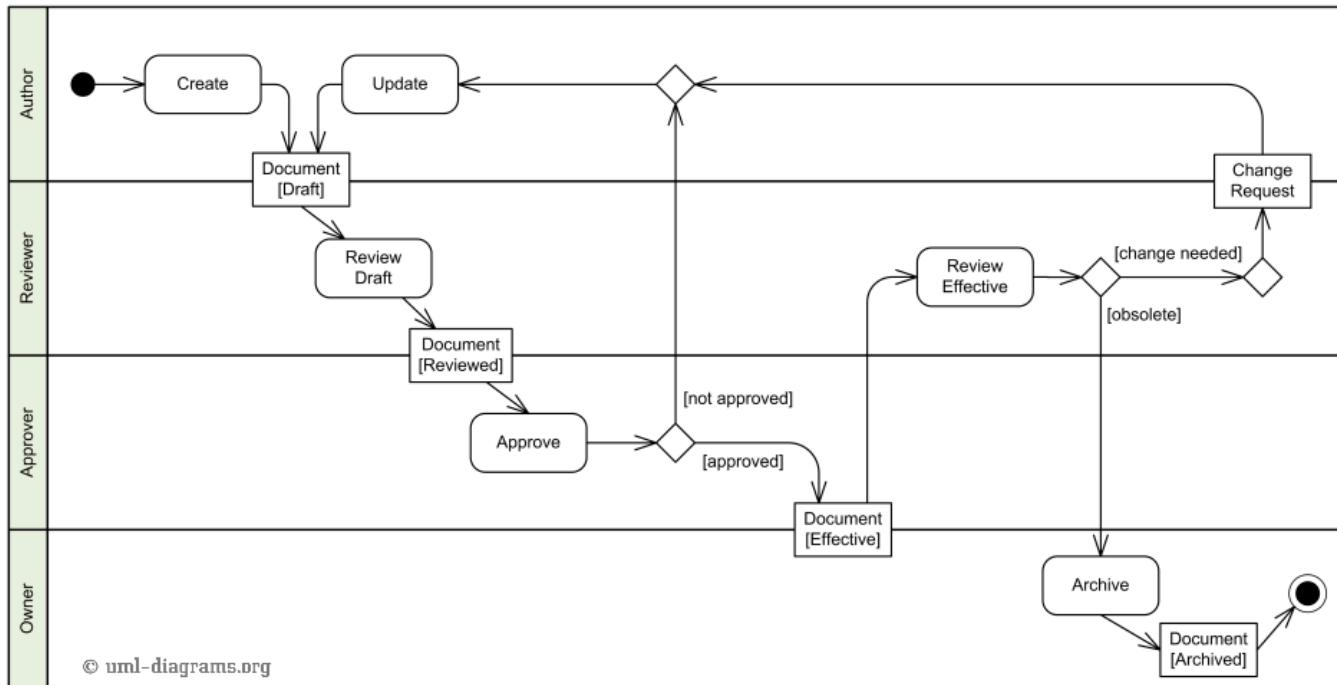




Activity diagram

Lanes may be vertical or horizontal.

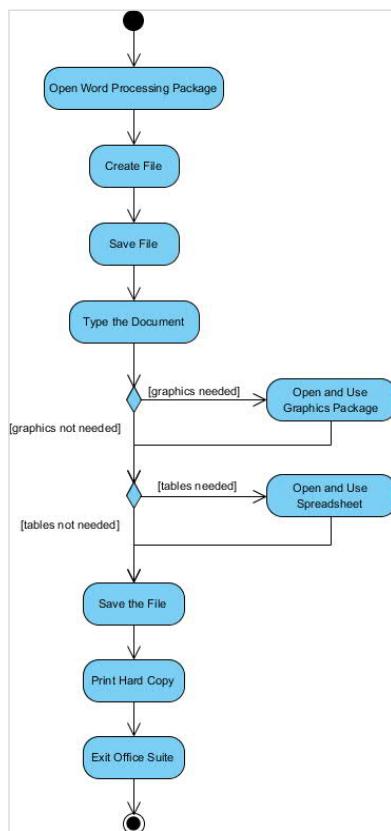




"Swimlane"; this activity diagram shows responsibilities of different roles and flow or sequence of document changes. Alternative type of diagram - state machine diagram - could also be used in this case to show how document changes its state over time.

Activity diagram Word processor

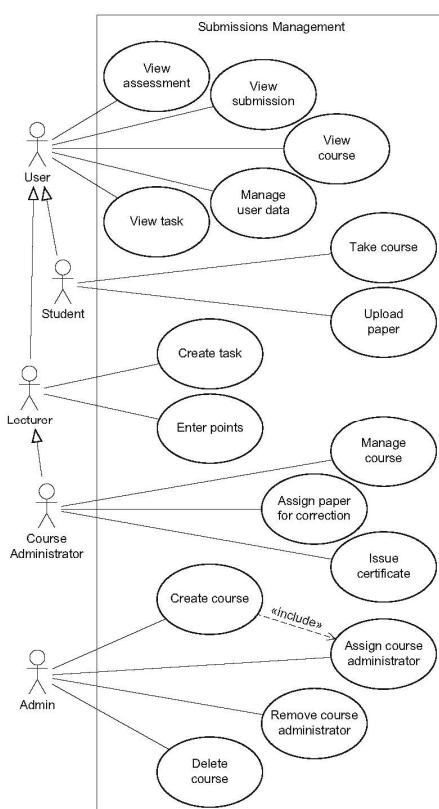
[<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/>]



Example of three diagrams

student
administration
system of a
university,
Use Case
diagram

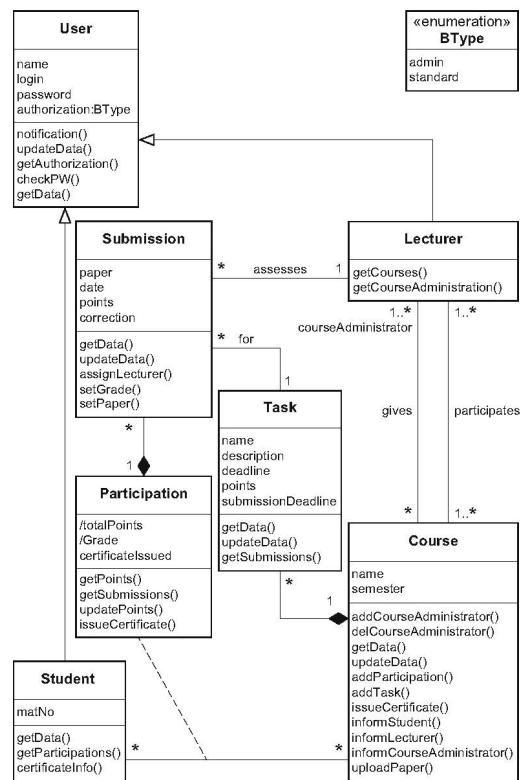
Figure 8.7
Use case diagram for a
submission system



student
administration
system of a
university,
Class diagram

[UML@Classroom, 2015]

Figure 8.8
Class diagram for a submission system



TUNI / COMP.SE.100-EN

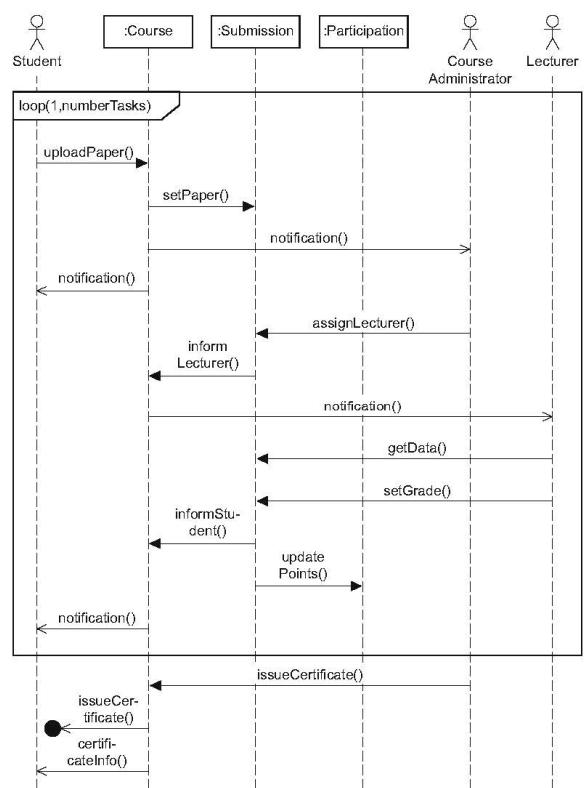
30.09.2020 10.03

55

student
administration
system of a
university,
Sequence
diagram

[UML@Classroom, 2015]

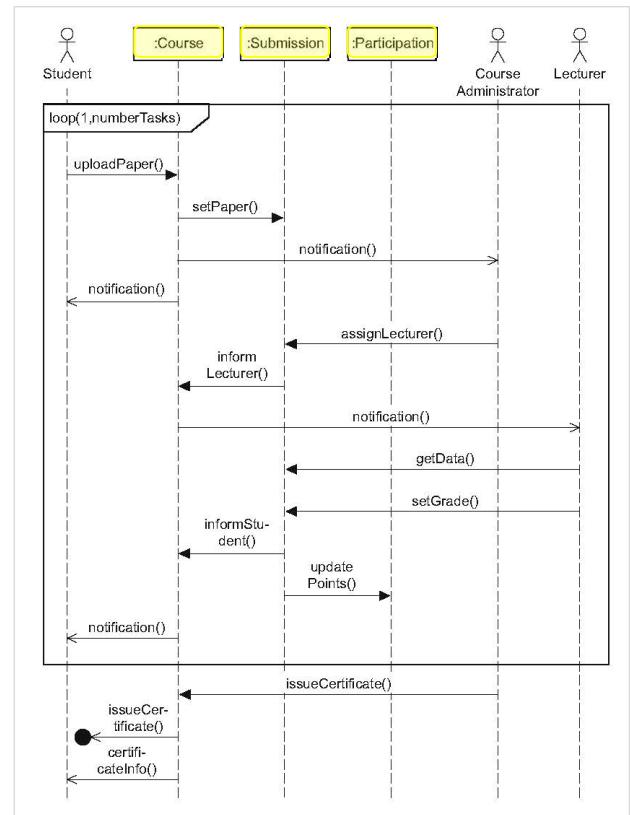
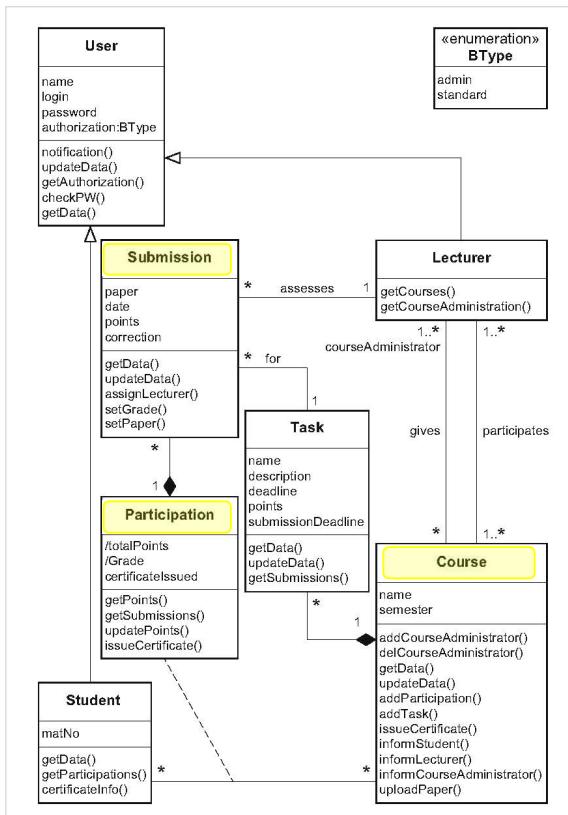
Figure 8.9
Communication flows



TUNI / COMP.SE.100-EN

30.09.2020 10.03

56



Flowcharts

Flowchart (FI: vuokaavio)

- Used in early years to describe software logic (after which pseudocode was written), at requirements specification phase
- Suits wherever logical function description is needed (e.g. algorithms or error searching).

The screenshot shows the ISO website homepage. At the top, the ISO logo and the text "International Organization for Standardization" are visible, along with the tagline "Great things happen when the world agrees". Below the header, there are navigation links for "Standards", "All about ISO", "Taking part", "Store" (which is highlighted in red), and a search bar. Under the "Store" link, there are categories for "Standards catalogue" and "Publications and products". The main content area displays the ISO 5807:1985 standard, with its title "ISO 5807:1985" in large bold letters, a "Preview" button, and a brief description: "Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts".

30.09.2020 10.03

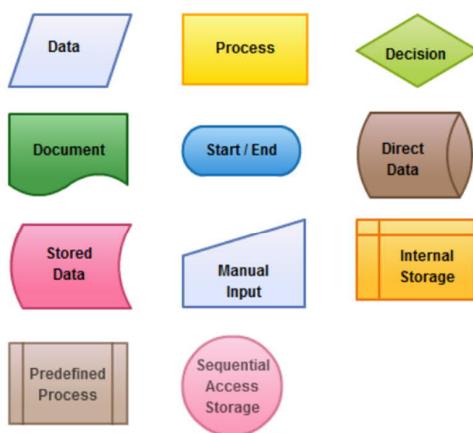
TUNI / COMP.SE.100-EN

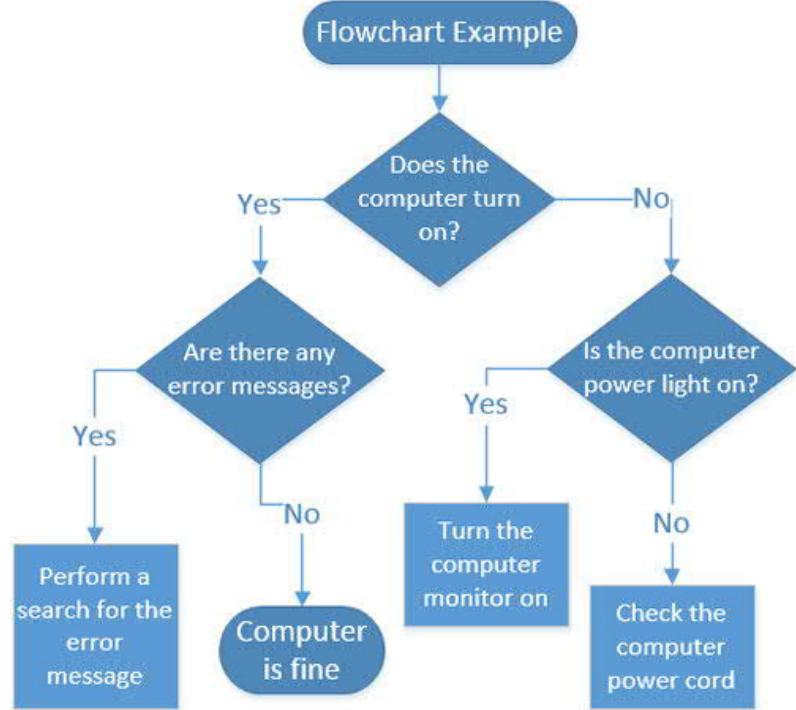
59

History of Flowcharts

Frank Gilberth introduced flowcharts in 1921, and they were called "Process Flow Charts" at the beginning. Allan H. Mogensen is credited with training business people on how to use flowcharts. Wikipedia has a great summary of the history of flowcharts, read more in [this wiki section](#).

Flowchart Symbols Meaning





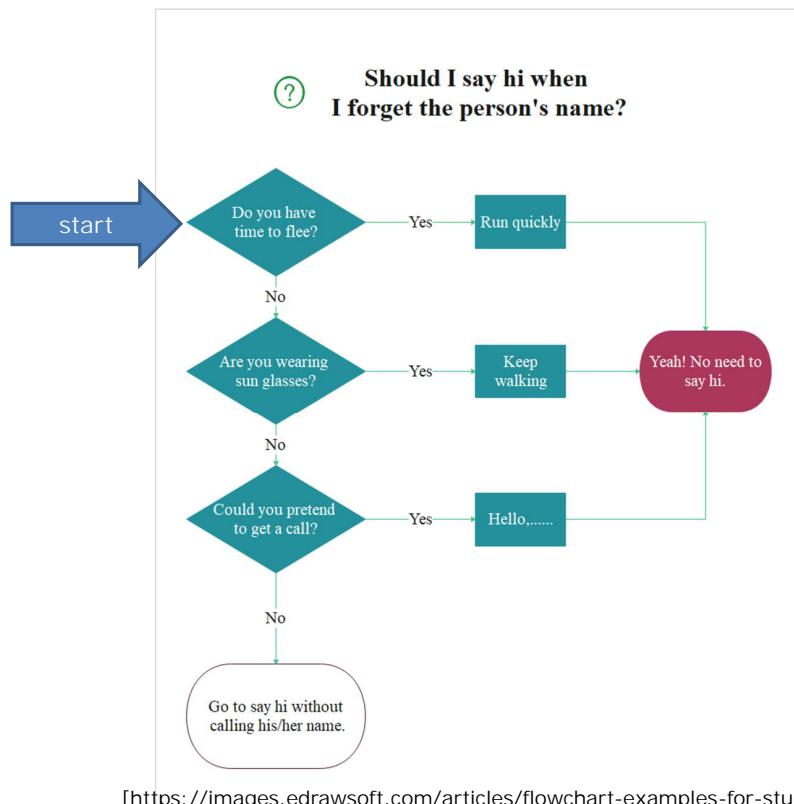
[www.computerhope.com/jargon/]

ComputerHope.com

30.09.2020 10.03

TUNI / COMP.SE.100-EN

61

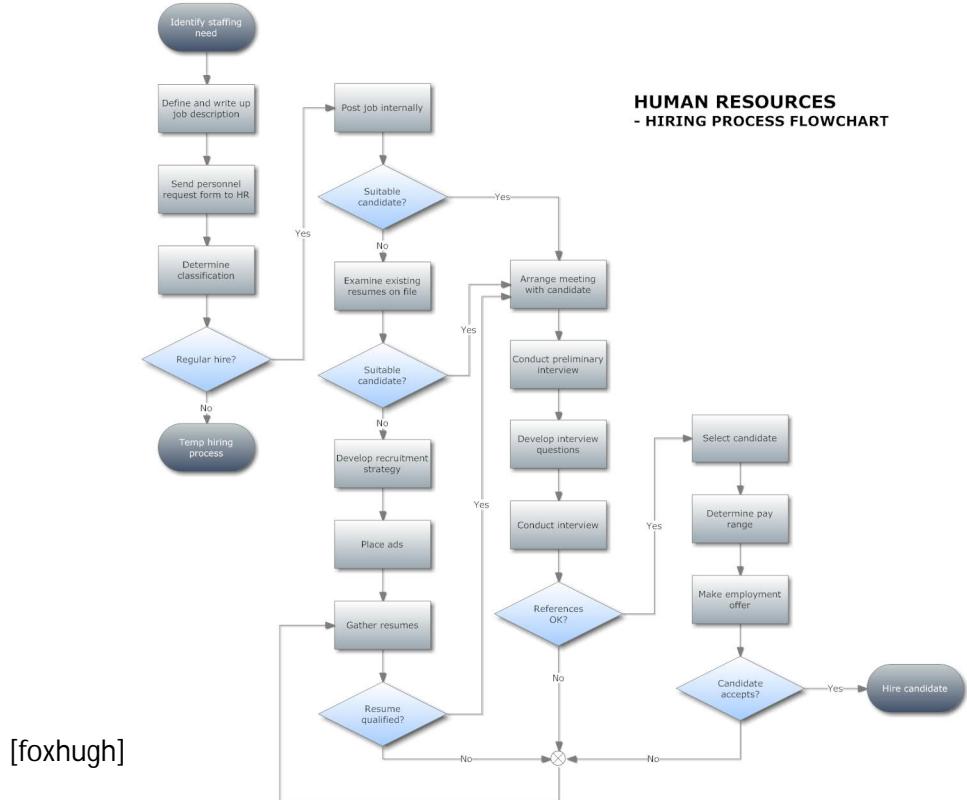


[<https://images.edrawsoft.com/articles/flowchart-examples-for-students/say-hi-flowchart.png>]

30.09.2020 11.14

TUNI / COMP.SE.100-EN

62



30.09.2020 10.03

63

Gantt diagram/chart

Gantt chart

Henry Laurence Gantt (1861-1919) was a mechanical engineer, management consultant and industry advisor. He developed Gantt charts in the second decade of the 20th century as a visual tool to show scheduled and actual progress of projects.

Accepted as a common-place project management tool today, it was quite a radical concept and an innovation of world-wide importance in the 1920s.

Gantt charts were first used on large construction projects like the Hoover Dam, started in 1931 and the interstate highway network which started in 1956.

Gantt chart, 2

Consulting Project



Gantt chart, 3



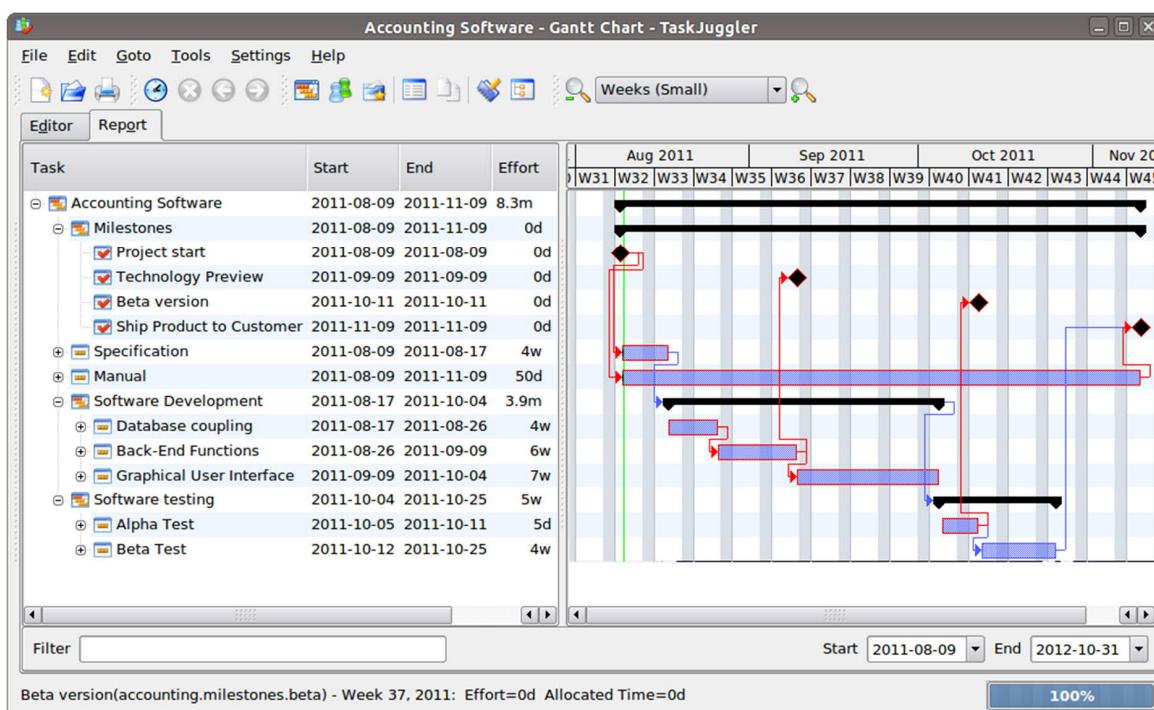
[www.conceptdraw.com]

30.09.2020 10.03

TUNI / COMP.SE.100-EN

67

Gantt chart, 4



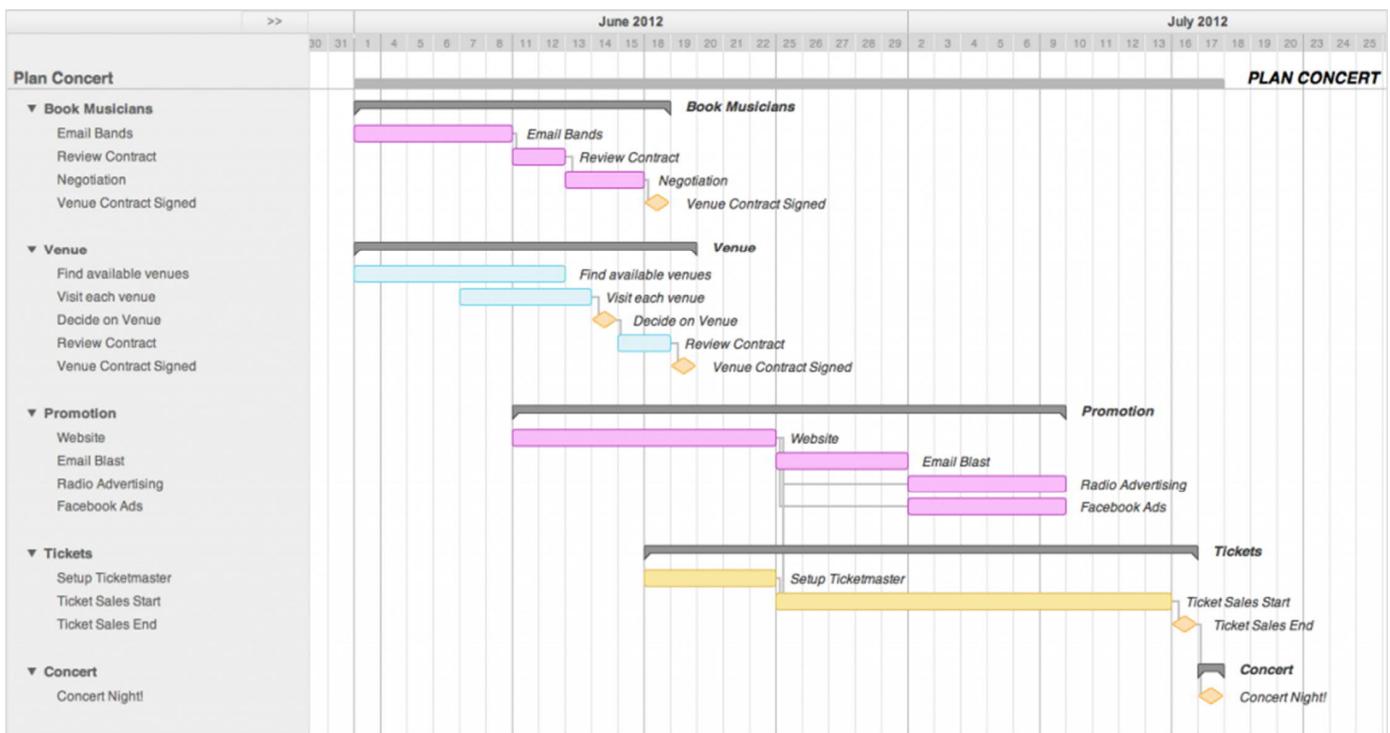
30.09.2020 10.03

TUNI / COMP.SE.100-EN

[http://orgmode.org]

68

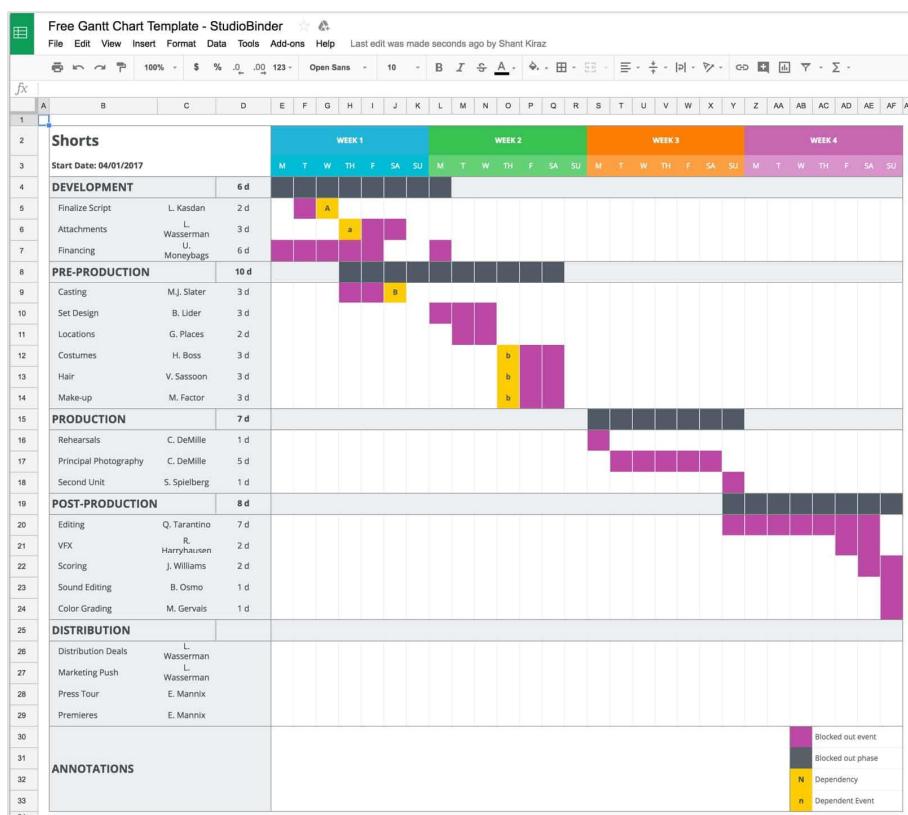
[<https://project-management.com/>]



TUNI / COMP.SE.100-EN

30.09.2020 11.43

69



[<https://s.studiobinder.com/>]

TUNI / COMP.SE.100-EN

30.09.2020 11.46

70

Highlights - What to remember

- state transitions are one important diagram in software development, it defines in which order actions may/should happen
- there are several different diagrams (techniques, styles), and terminology varies also
- drawing/diagramming symbols may vary according to the tool used
- the main point is to understand (read) different kind of state machines (and in UML behavior diagrams general)
- there are many many more diagrams and hundreds of details for special cases at UML specification
- remember the availability of Quick Reference Cards/Guides and "Cheat Sheets".

Now the additional L5 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Now the additional L5 extra slides are here

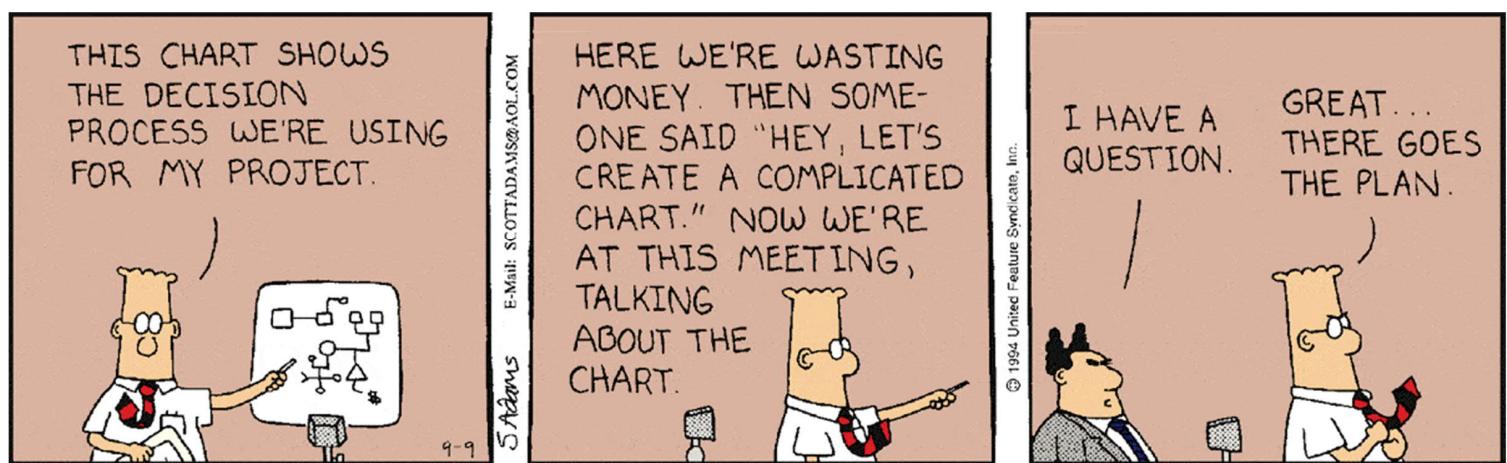
No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Now the additional L5 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.

Now the additional L5 extra slides are here

No time to show these at lectures, but otherwise good to know, at least if you are a major reader.



State transition diagrams = State machines

State machine [UML specification, 2015]

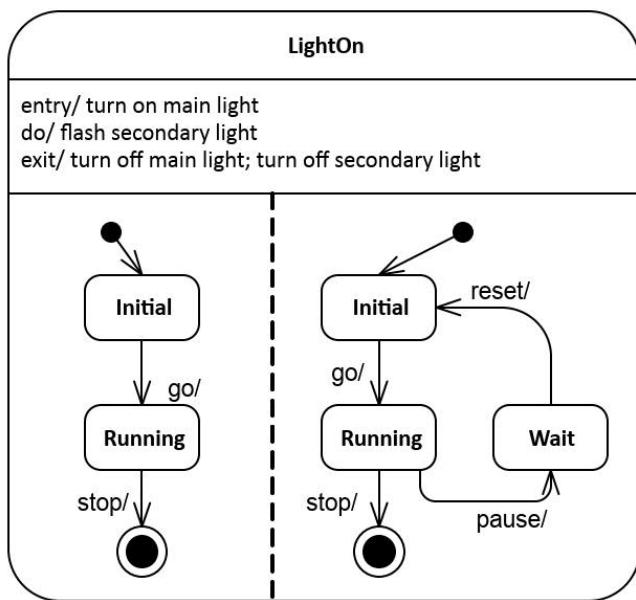
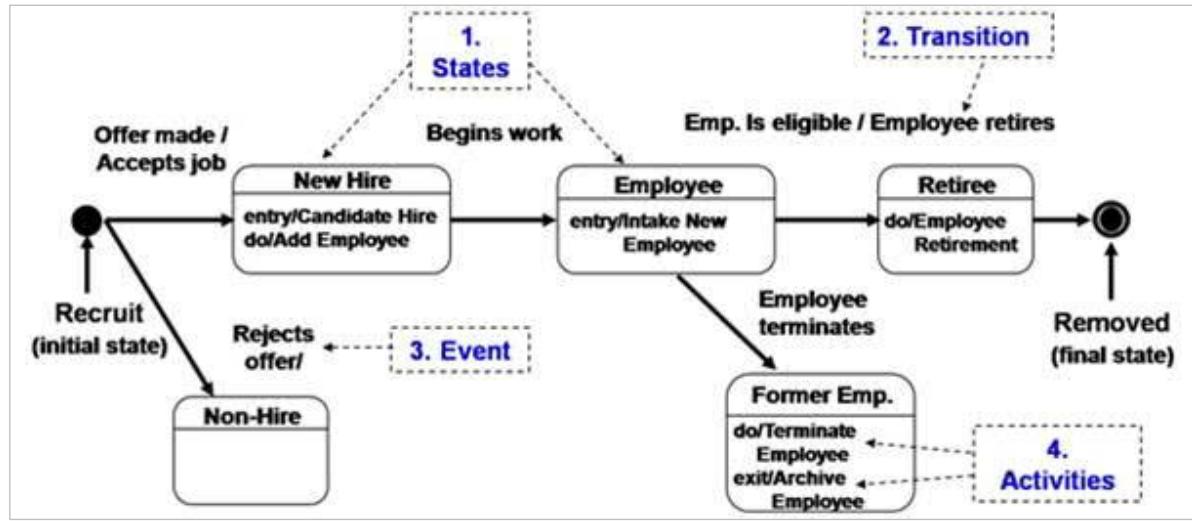


Figure 14.10 Composite State with two Regions and entry, exit, and do Behaviors



State diagram. It shows 1) Five example states for a Human Resources application, 2) Transitions that reflect the business process in moving from state to state, 3) The event or trigger that causes a move to a new state, and 4) Activities that can take place upon entering or exiting a given state.

[www.watermarklearning.com]

TUNI / COMP.SE.100-EN

30.09.2020 10.03

79

State machine [UML specification, 2015]

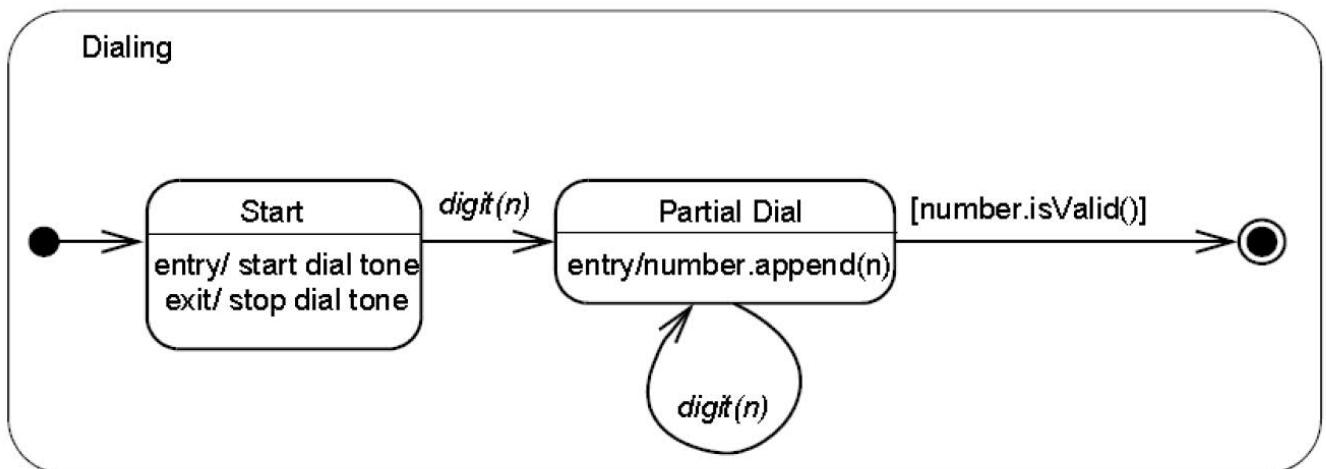
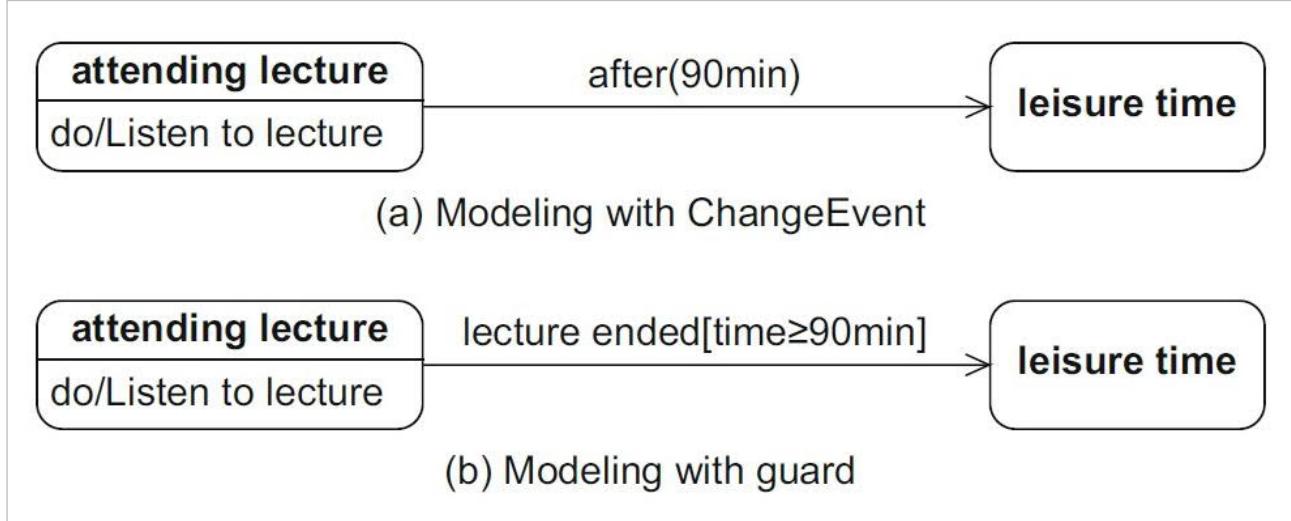
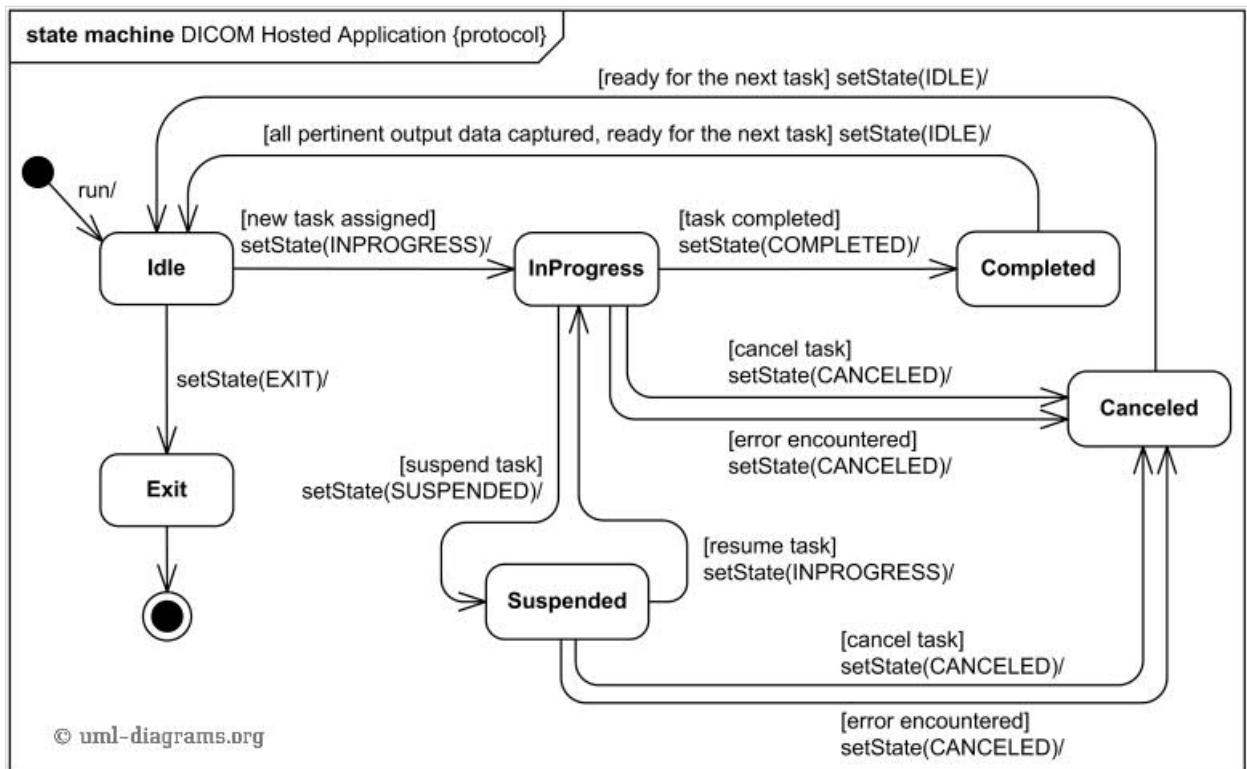


Figure 14.7 Composite State with two States

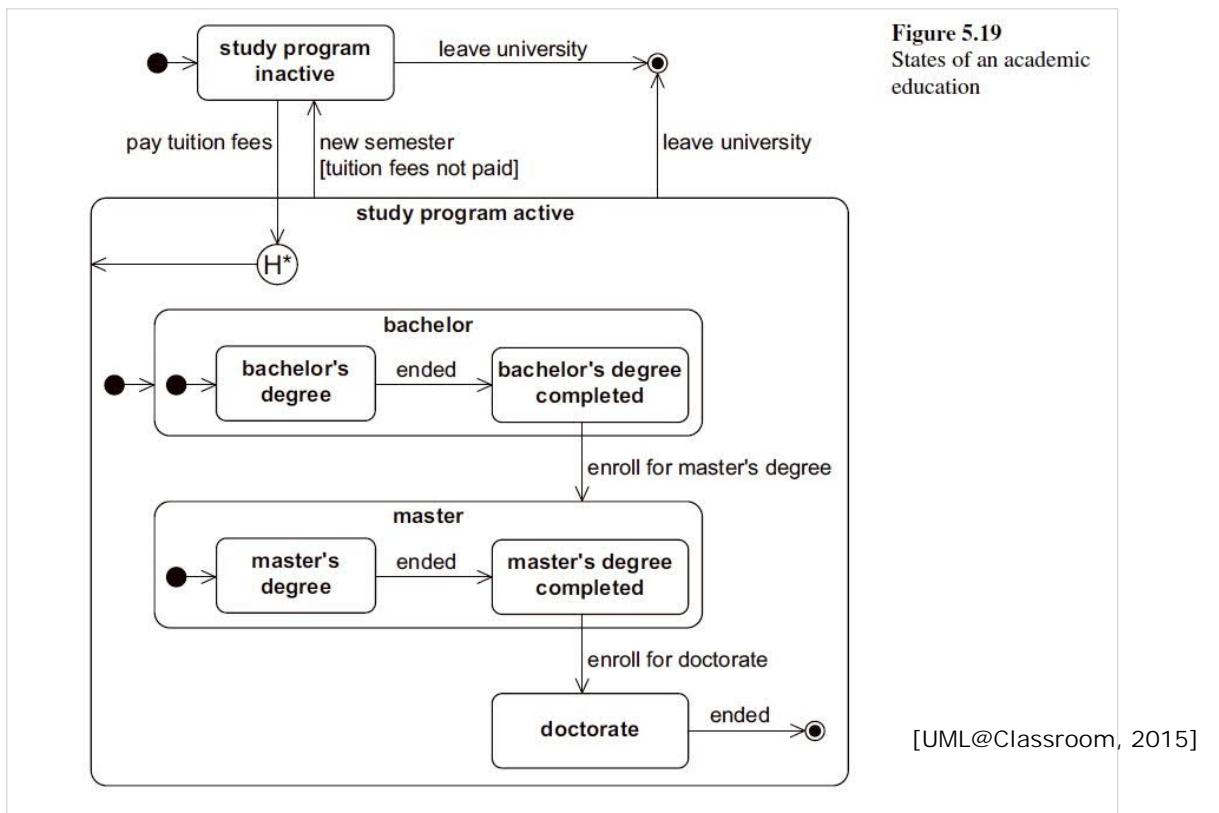


- a) student has leisure time after 90 minutes at any case.
- b) student has leisure time after lecture end, if the lecture has lasted at least 90 minutes.

A [guard] can never trigger an event itself.

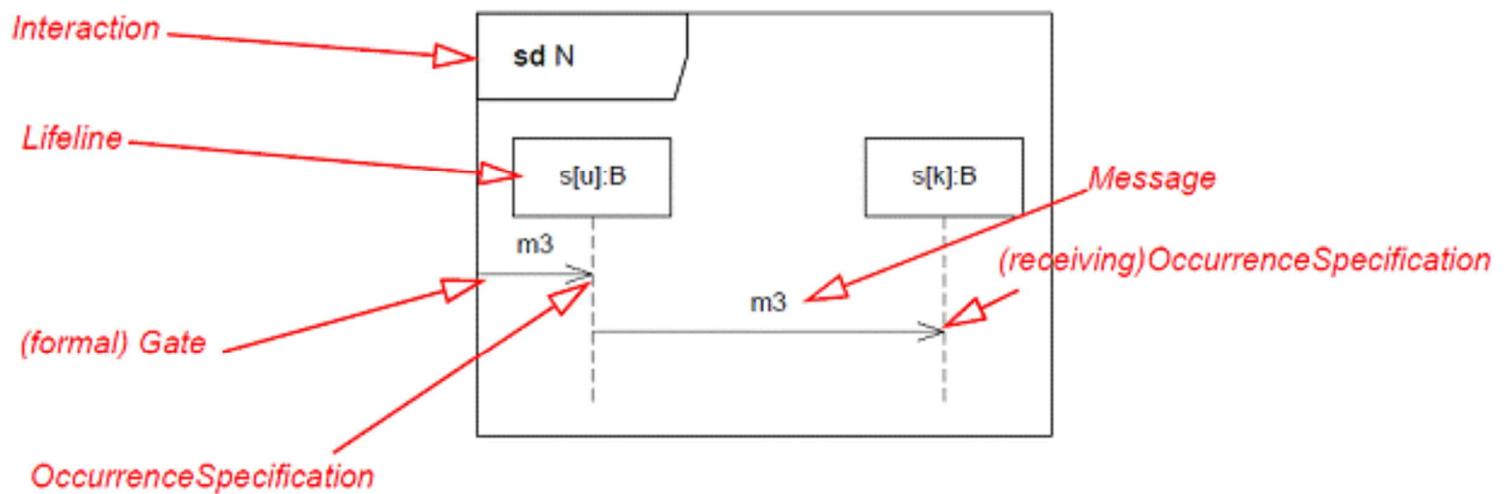
[Figure 5.11, UML@Classroom, 2015]

State machine



Sequence diagrams

Sequence diagram

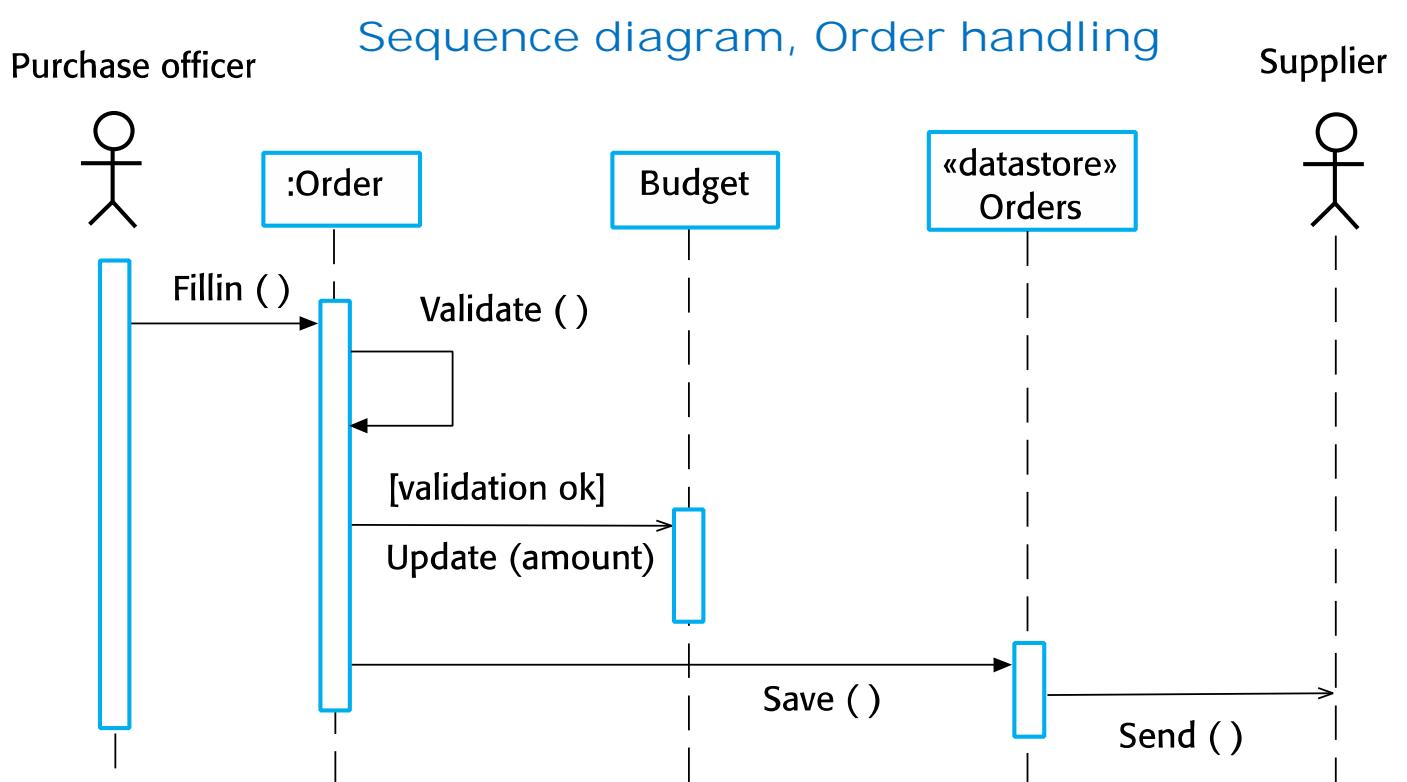


[UML 2.5 guide]

30.09.2020 10.03

TUNI / COMP.SE.100-EN

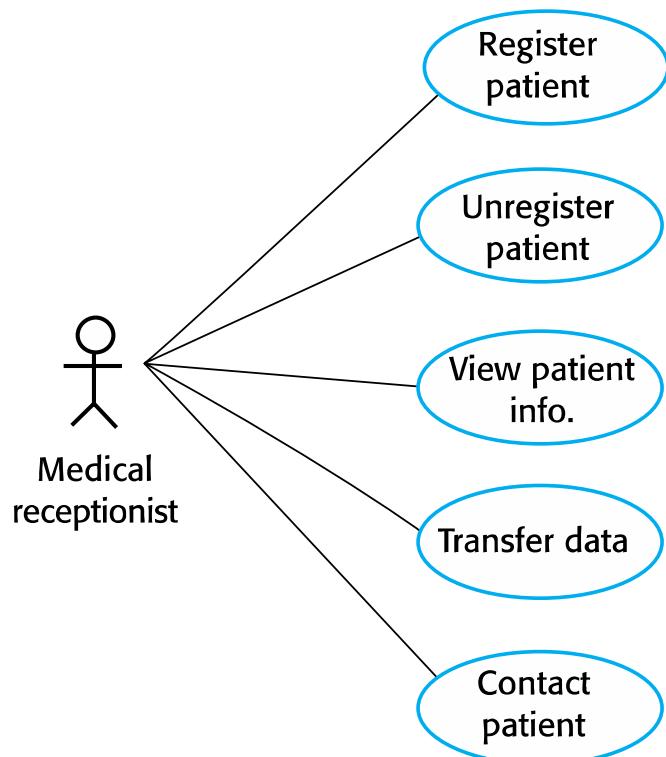
85



30.09.2020 10.03

TUNI / COMP.SE.100-EN

86



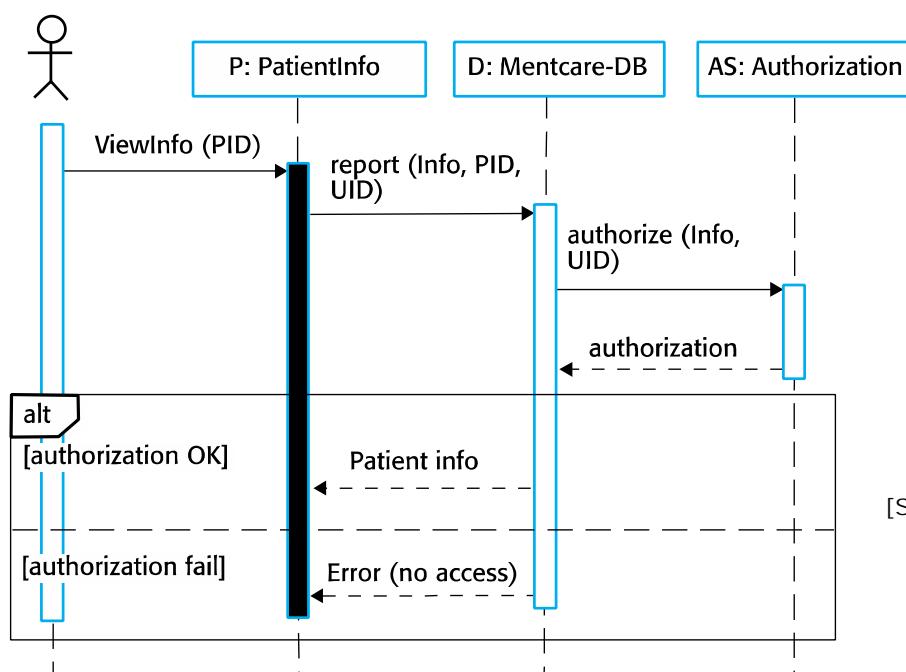
30.09.2020 10.03

TUNI / COMP.SE.100-EN

87

Sequence diagram, View patent information

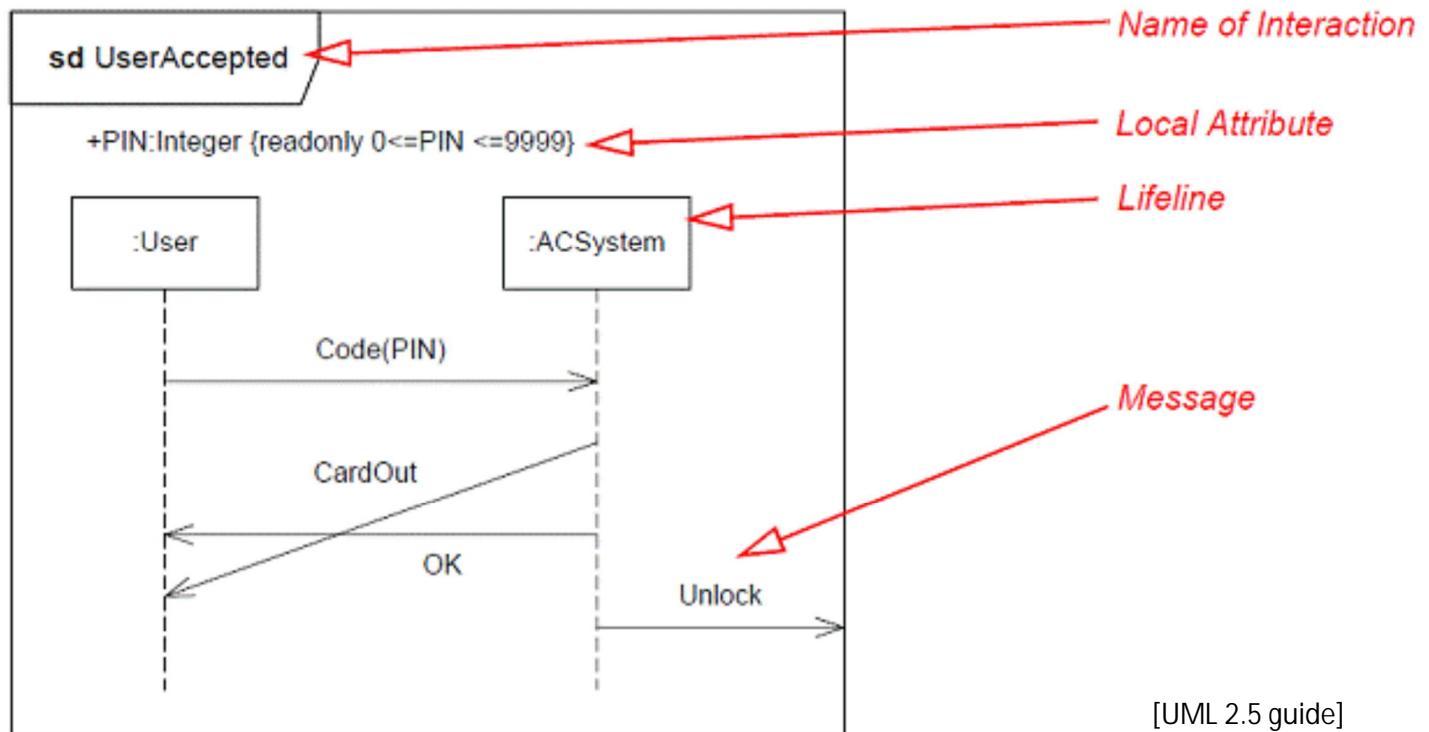
Medical Receptionist



30.09.2020 10.03

TUNI / COMP.SE.100-EN

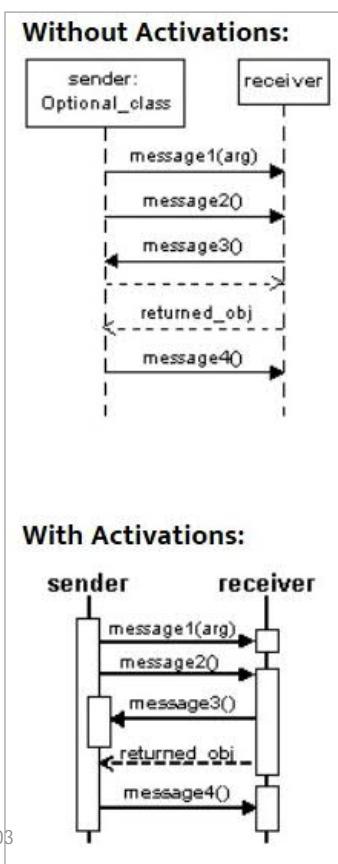
88



30.09.2020 10.03

TUNI / COMP.SE.100-EN

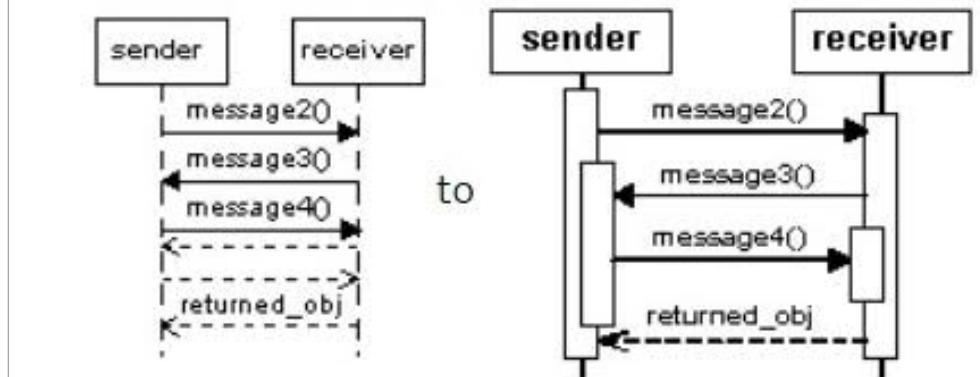
89



"lifelines" may have activations "pipes" which describe when event is alive

(FI: "elämänlinjassa" voi olla "putkia" jotka kuvaavat tapahtuman kestoa (loppumista)).

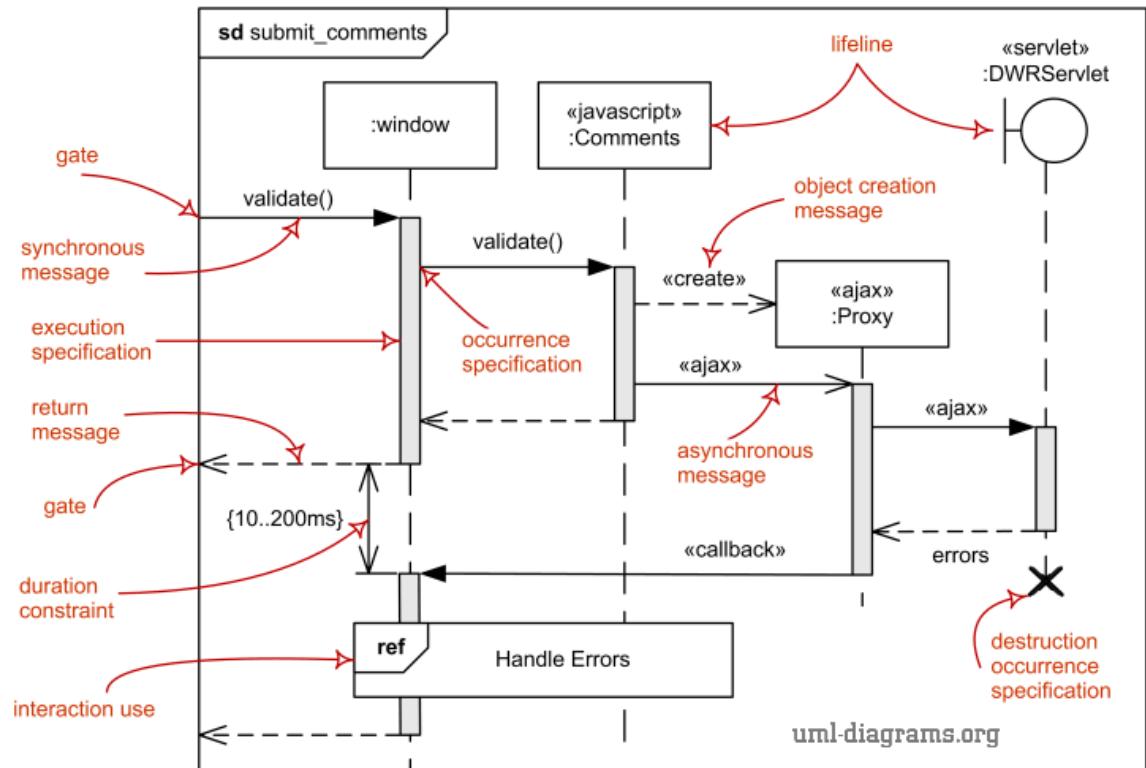
- Activiations are optional, but much easier to read.



30.09.2020 10.03

TUNI / COMP.SE.100-EN

90

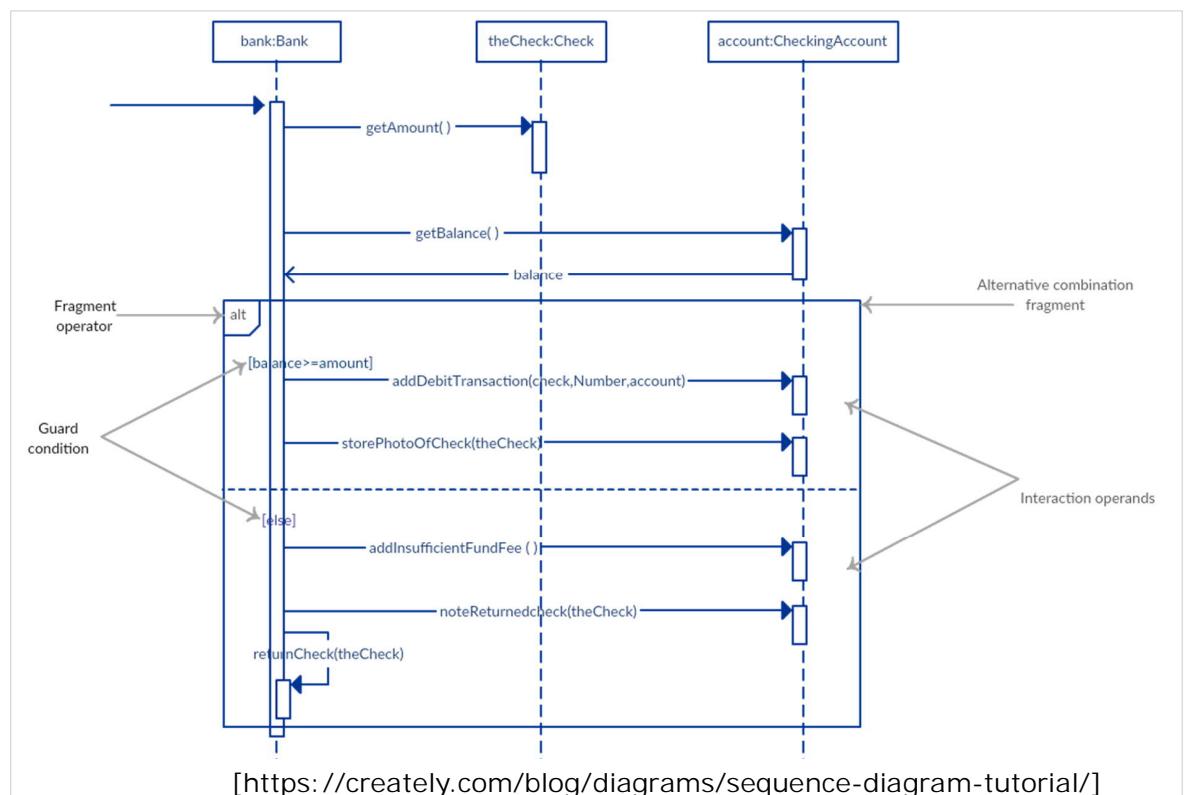


30.09.2020 10.03

TUNI / COMP.SE.100-EN

[www.uml-diagrams.org]

91



[https://creately.com/blog/diagrams/sequence-diagram-tutorial/]

30.09.2020 13.25

TUNI / COMP.SE.100-EN

92

Facebook user authentication (UML sequence diagram example)

- An example of **UML sequence diagram** which shows how Facebook (FB) user could be authenticated in a web application to allow access to his/her FB resources. Facebook uses OAuth 2.0 protocol framework which enables web application (called "client"), which is usually not the FB resource owner but is acting on the FB user's behalf, to request access to resources controlled by the FB user and hosted by the FB server. Instead of using the FB user credentials to access protected resources, the web application obtains an access token.
- Web application should be registered by Facebook to have an application ID (client_id) and secret (client_secret). When request to some protected Facebook resources is received, web browser ("user agent") is redirected to Facebook's authorization server with application ID and the URL the user should be redirected back to after the authorization process.
- User receives back Request for Permission form. If the user authorizes the application to get his/her data, Facebook authorization server redirects back to the URI that was specified before together with authorization code ("verification string"). The authorization code can be exchanged by web application for an OAuth access token.
- If web application obtains the access token for a FB user, it can perform authorized requests on behalf of that FB user by including the access token in the Facebook Graph API requests. If the user did not authorize web application, Facebook issues redirect request to the URI specified before, and adds the error_reason parameter to notify the web application that authorization request was denied.

[<http://www.uml-diagrams.org/facebook-authentication-uml-sequence-diagram-example.html>]

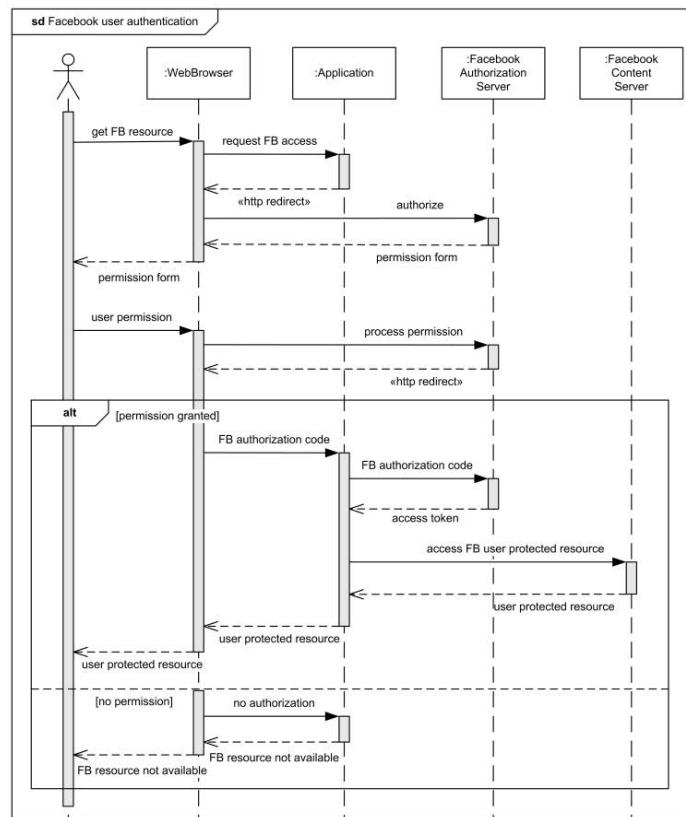
30.09.2020 10.03

TUNI / COMP.SE.100-EN

93

Facebook user authentication (UML sequence diagram example)

[<http://www.uml-diagrams.org/facebook-authentication-uml-sequence-diagram-example.html>]



30.09.2020 10.03

TUNI / COMP.SE.100-EN

94

Example: Submit comments to Pluck using DWR, AJAX, JSON

- An example of **sequence diagram** which shows how user comments on some article are submitted to Pluck using various AJAX technologies. (See example of interaction overview diagram with similar semantics at [Submit Comments to Pluck using DWR, AJAX, JSON.](#).)
- Comments submitted by user are first validated by the web site hosting articles. DWR technology (AJAX for Java) is used to convert user comments HTML form data into Java object and possible validation errors - back into JavaScript callbacks for errors.
- Comments that look Ok are submitted to Pluck server hosting all comments to all articles. AJAX technology is also used in this case both to submit the new comment as well as to get back the list of all recent comments (including the new one). JSON is used to get comments back.
- This diagram also shows some duration constraints. (Disclaimer: constraints shown are fictitious and not describing any real communication.) For example, according to the diagram callback waiting time for posted comment varies from 1 to 4 seconds. At the same time, requesting all posted comments takes only up to 100 msec.

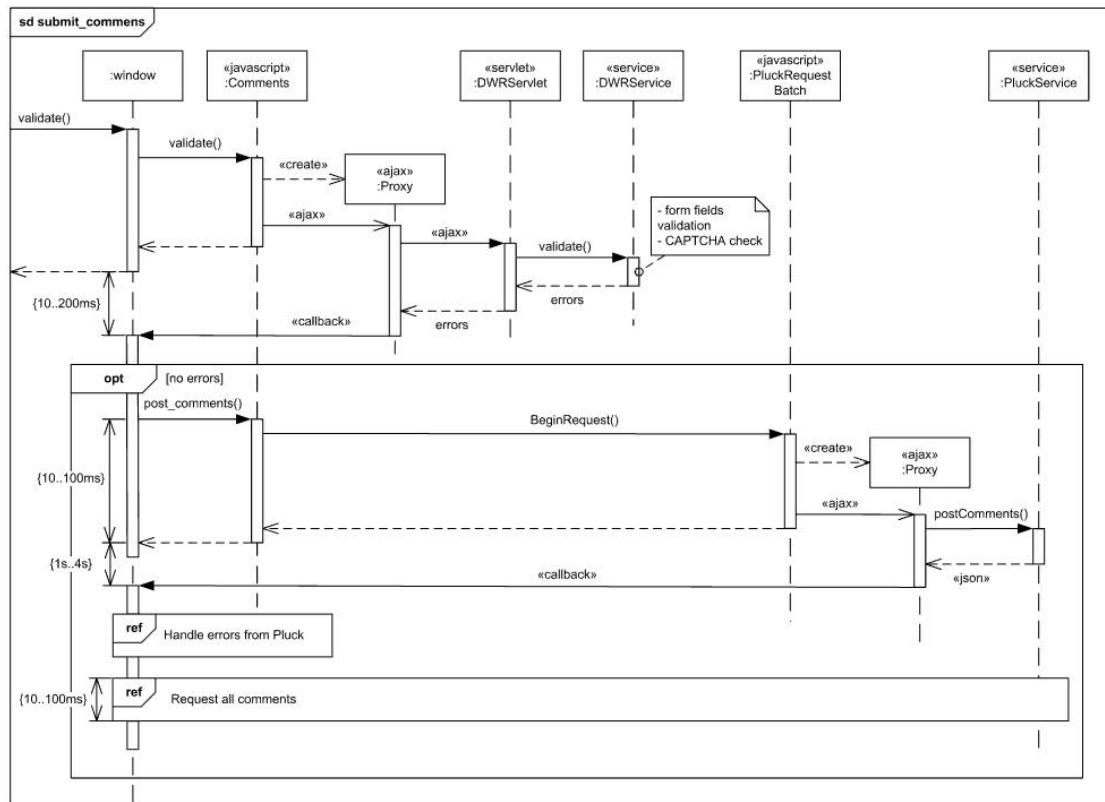
[<http://www.uml-diagrams.org/pluck-comments-uml-sequence-diagram-example.html>]

30.09.2020 10.03

TUNI / COMP.SE.100-EN

95

Example:
Submit
comments
to Pluck
using DWR,
AJAX, JSON

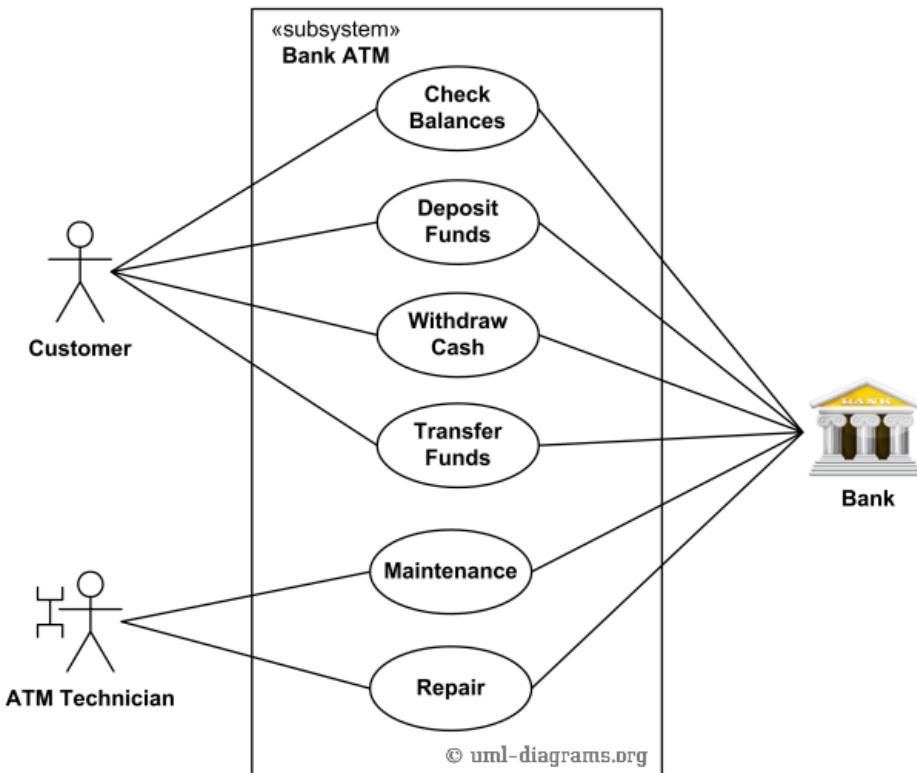


30.09.2020 10.03

TUNI / COMP.SE.100-EN

96

An example of use case diagram for Bank ATM subsystem - top level use cases



30.09.2020 10.03

TUNI / COMP.SE.100-EN

97

Bank account UML class diagram example

- This is an example describing some types of Bank Accounts using UML generalization sets.
- Bank accounts could be grouped into UML generalization sets based on different criteria. Example diagram below shows bank accounts split by liability type and account type. These two orthogonal dimensions also have corresponding power types - LiabilityType and AccountType.
- Bank account could be used either for personal or for business purposes. To show that it assumes complete coverage and there is no overlapping, we have liability type constraints shown as { complete, disjoint } .
- Note, that business owners still may use a personal bank account for their business purposes but it is not recommended primarily because it can affect legal liability of business owner. From the bank's point of view, e.g. when opening an account, these two are two different kinds of accounts.
- Another classification of bank accounts is based on related options and features and is shown below as account type generalization set. To show that it this set is incomplete but still there is no overlapping, we have account type constraints shown as { incomplete, disjoint } .
- Note, that it is possible to have bank accounts with different combinations of account liability and account type, for example, personal savings account or business money market account.

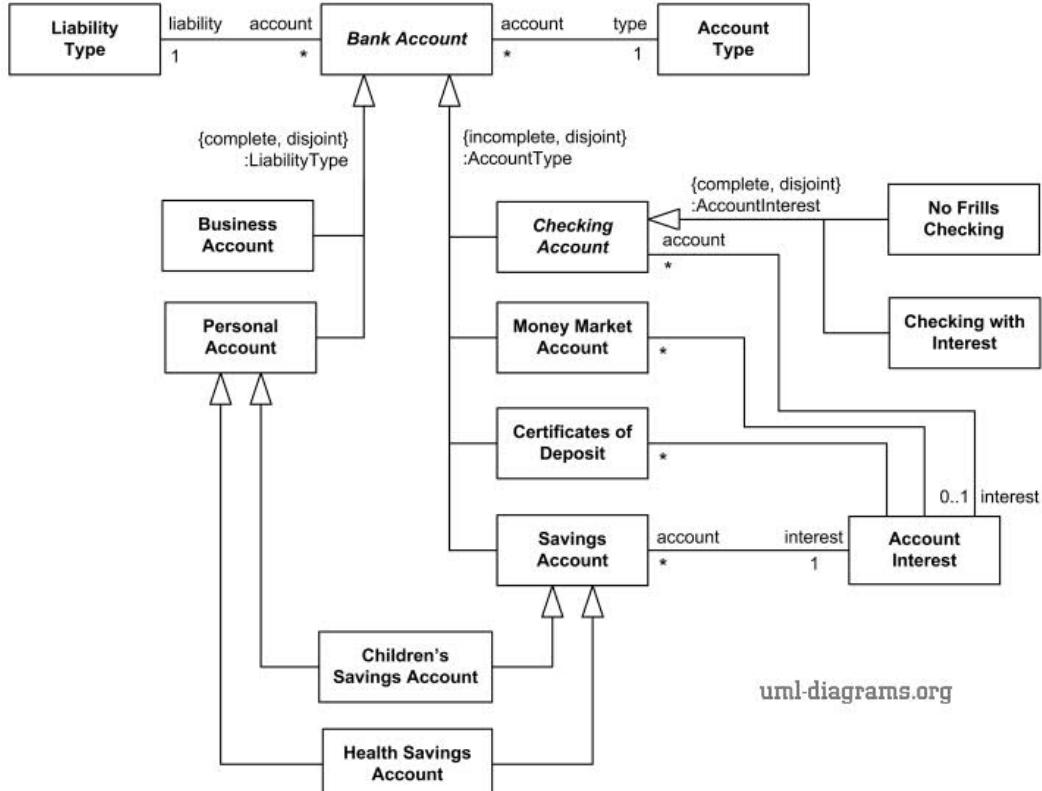
[<http://www.uml-diagrams.org/examples/bank-account-domain-diagram-example.html>]

30.09.2020 10.03

TUNI / COMP.SE.100-EN

98

Bank account UML class diagram example



30.09.2020 10.03

TUNI / COMP.SE.100-EN

99

Additional material about UML

There are hundreds of web pages and YouTube videos about UML diagrams.

- <http://www.omg.org/spec/UML/2.5/> (OMG:n spesifikaatio)
- <http://www.uml.org/>
- <http://www.uml-diagrams.org/>
- <https://www.tutorialspoint.com/uml/>
- <https://modeling-languages.com/best-uml-cheatsheets-and-reference-guides/>
- <https://modeling-languages.com/list-uml-books/>

Videoita YouTubessa, mm.

[Derek Banas, 9 videos, 2012], for example

- "UML 2.0 Tutorial" (and Use Case)
- "UML 2.0 Class Diagrams" (programming...)
- "UML 2 Sequence Diagrams".

30.09.2020 10.03

TUNI / COMP.SE.100-EN

100

Activity diagrams

Activity diagram [UML specification, 2015]

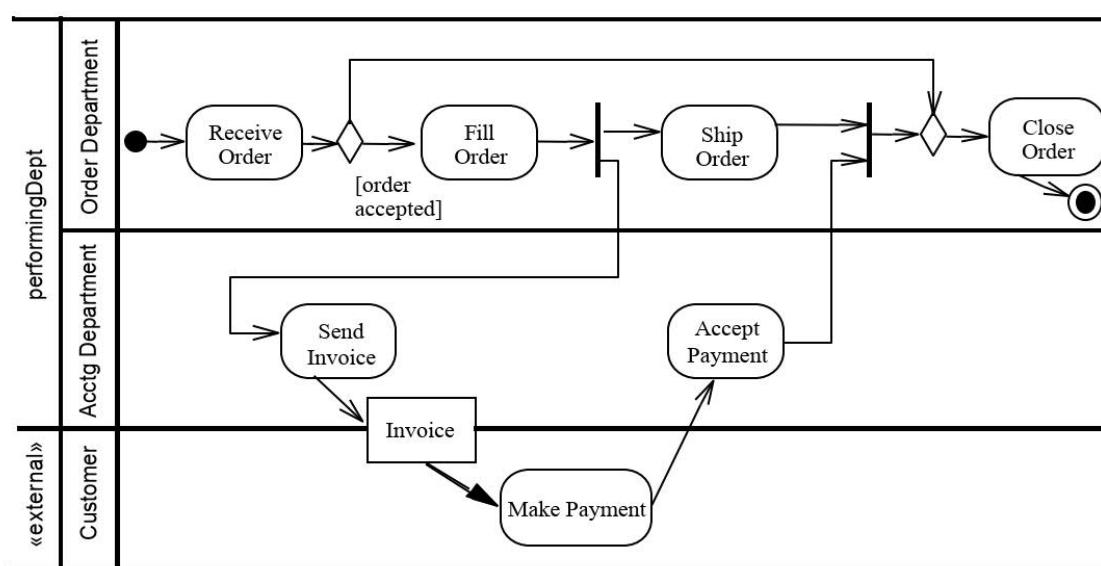


Figure 15.70 ActivityPartitions using swimlane notation

Activity diagram [UML specification, 2015]

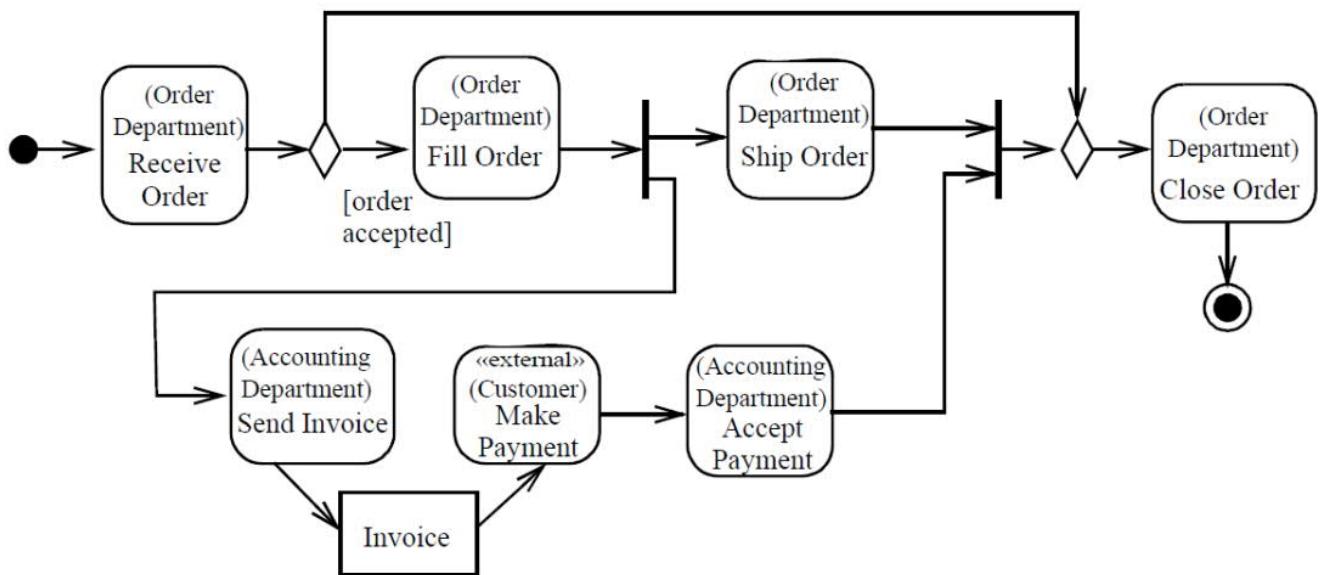
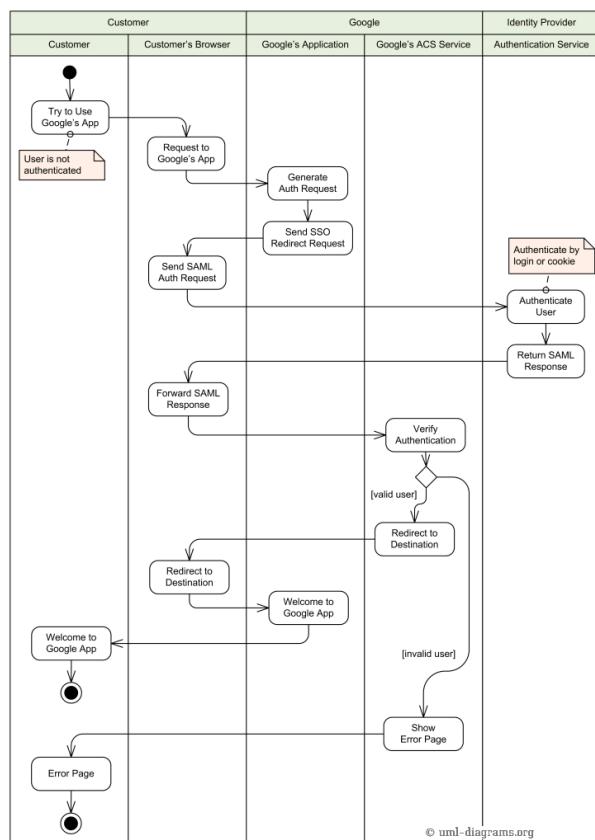
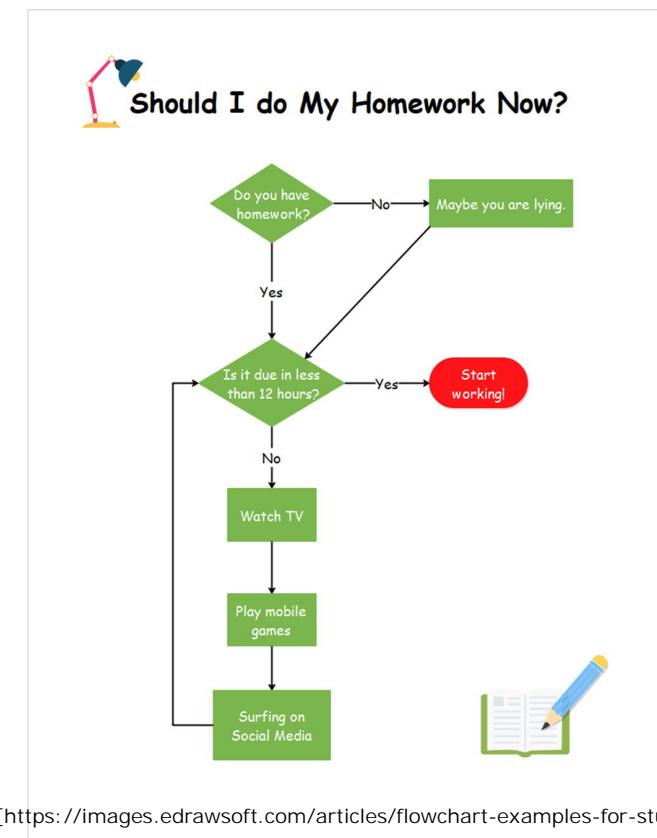


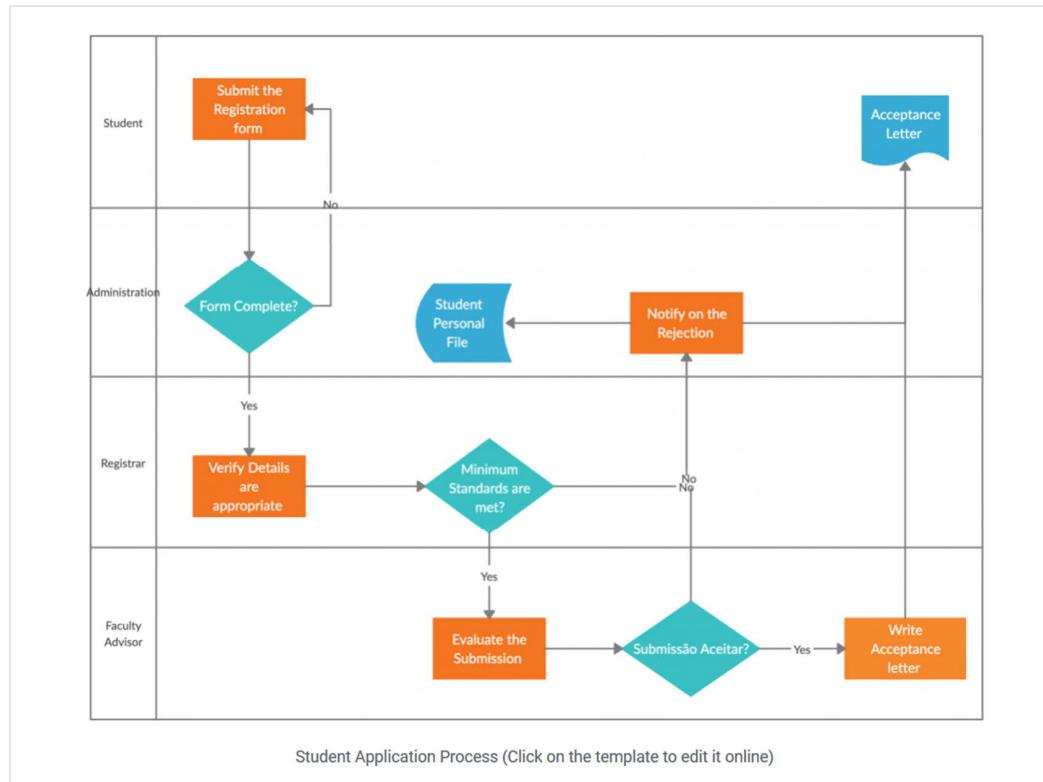
Figure 15.71 ActivityPartitions using annotation

Google apps SSO
single-sign-on



Flowcharts





[<https://creately.com/blog/diagrams/flowchart-guide-flowchart-tutorial/>]

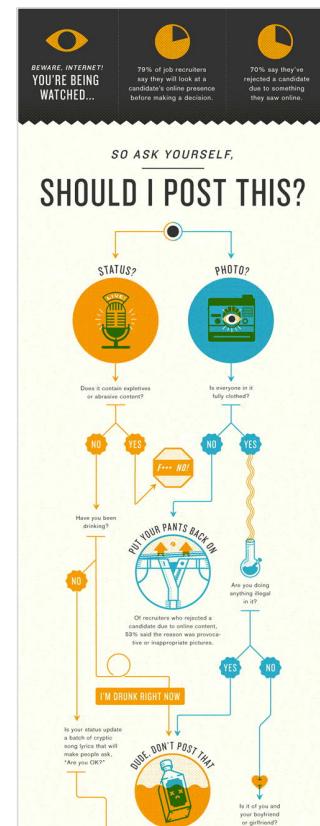
TUNI / COMP.SE.100-EN

30.09.2020 11.23 107



[<https://visme.co/blog/wp-content/uploads/2017/07/21-Creative-Flowchart-Examples-for-Making-Important-Life-Decisions-Include-Statistics.png>]

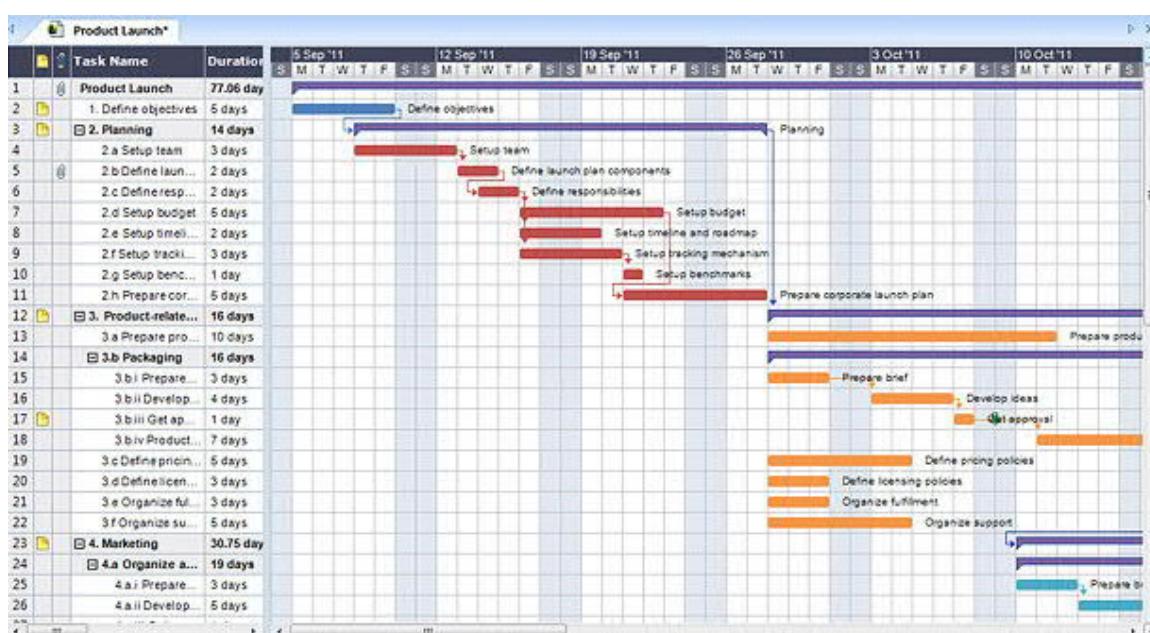
TUNI / COMP.SE.100-EN



30.09.2020 11.26 108

Gantt diagram/chart

Gantt chart, project time line



[<http://www.matchware.com>]



Now the additional L5 extra slides set ends here

Now the additional L5
extra slides set ends here